异步图书
www.epubit.com

The Hundred-Page Machine Learning Book

机器学习精讲

[加拿大] 安德烈·布可夫（Andriy Burkov）著

韩江雷 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

ISBN：978-7-115-51853-8



[illegible] [[illegible]] [illegible]·[illegible]Andriy Burkov[illegible]

[illegible] [illegible]

[illegible] [illegible]

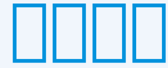
[illegible] [illegible]11[illegible]

[illegible] 100164 [illegible] 315@ptpress.com.cn

[illegible] http://www.ptpress.com.cn

[illegible](010)81055410

[illegible](010)81055315



[illegible]

[illegible] 11 [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]“[illegible]”[illegible]5[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible] “[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

2018[illegible]“[illegible]”[illegible]

[illegible]

[illegible]

2019



[illegible]——[illegible]“[illegible]”[illegible]

[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]·[illegible](Arther Samuel)[illegible]1959[illegible]IBM[illegible]IBM[illegible]2010[illegible]“[illegible]”(Cognitive Computing)[illegible]

[illegible]“[illegible]”[illegible]——[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]



第1章 [illegible]

1.1 [illegible]

[illegible],[illegible]
[illegible],[illegible],[illegible]。

[illegible],[illegible],[illegible]
[illegible]。

[illegible],[illegible]“[illegible]”[illegible]“[illegible]”[illegible]。

1.2 [illegible]

[illegible]4[illegible]:[illegible](supervised)[illegible](semi-supervised)[illegible](unsupervised)[illegible](reinforcement)[illegible]。

1.2.1 [illegible]

[illegible](supervised learning)[1][illegible](dataset),[illegible](labeled example)[2],[illegible]$\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$[illegible]$N$[illegible],[illegible]$\boldsymbol{x}_i$[illegible](feature vector)[illegible]$j=1,\ldots,D$[illegible](feature),[illegible]$x^{(j)}$[illegible],[illegible]x[illegible],[illegible]

$x^{(1)}$[illegible],[illegible]$x^{(2)}$[illegible],[illegible]$x^{(3)}$[illegible],[illegible],[illegible]j[illegible]$\boldsymbol{x}_i$[illegible]$x_i^{(2)}$[illegible]([illegible]),[illegible],[illegible](label)y_i[illegible](class)[illegible]{1,2,…,C},[illegible],[illegible],[illegible],[illegible][3][illegible],[illegible],[illegible]——{“[illegible]”,“[illegible]”}[illegible]

[illegible](supervised learning algorithm)[illegible],[illegible],[illegible],[illegible]

1.2.2 [illegible]

[illegible],[illegible](unsupervised learning)[illegible],[illegible]$\{(\boldsymbol{x}_i)\}_{i=1}^N$[illegible](unsupervised learning algorithm)[illegible](model)[illegible]$\boldsymbol{x}$[illegible],[illegible]:[illegible](clustering)[illegible](cluster)[illegible];[illegible](dimensionality reduction)[illegible]$\boldsymbol{x}$[illegible];[illegible](outlier detection)[illegible],[illegible]$\boldsymbol{x}$[illegible]“[illegible]”[illegible]

1.2.3 [illegible]

[illegible](semi-supervised learning)[illegible],[illegible](semi-supervised learning algorithm)[illegible],[illegible],[illegible]

[illegible],[illegible]?[illegible],[illegible],[illegible]:[illegible],[illegible]

1.2.4 [illegible]

[illegible](reinforcement learning)[illegible],[illegible]“[illegible]”[illegible],[illegible](state)[illegible],[illegible],[illegible](policy)[illegible]



[illegible],[illegible],[illegible],[illegible]“[illegible]”[illegible]

□□□□□□□□□□□□□□,□□□□□□□□□,□□,□□□□□□□□□□□□□□□□□□□□

1.3 □□□□□□□□□

□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□(□□,□□)□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□(□□,“□□□□”“□□□□□”“□”“□”“□□”□)□□□□□□,□□□□□□□(□□,□□□□□□□□□□□4□□□)□□□[□□“□□□”□□□□□(“□□”“□□”“□□”)]□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□“□□□□”□“□□□□□”(□□□□□,□□□□□□□□)□□□□,□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□(□□□□□),□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□”(bag of words)□□□□□,□□□□□3 500□□□□□□[4](□□□□□)□□□□□□□□□□□

- □□□□□□□□□□“□”,□□□□□□□□□1;□□□0□
- □□□□□□□□“□”,□□□□□□□□□1;□□□0□
- □□□□□□□□“□”,□□□□□□□□□1;□□□0□

□□□□□□,□□□□□□□□□□□□□□3 500□□□□□,□□□□□□1□0□

□□,□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□,□□□□0□1□□“□□□□□”□“□□□□”□□□,□□□□□□□□□(Support Vector Machine,SVM)□□□□□□□□□□□□□□□□+1(□□1)□□□□□(“□□□□”),□□□□□(“□□□□□”)□-1(□□1)□

□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□

SVM□□□□□□□□□□□□□□□□□□□□(□□□□□□,□□□3 500□)□□□□□□□□□3 500□□□□,□□□3 499□□□□(hyperplane)□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□(decision boundary)□

□□□□□□□□□□□□□□□□□□□□□□□,□□□□□***w***,□□□□□□***x***□□□□□□,□□□□□*b*:

$$\boldsymbol{w}\boldsymbol{x}-b=0$$

□□,***wx***□□□□$w^{(1)}x^{(1)}+w^{(2)}x^{(2)}+\ldots+w^{(D)}x^{(D)}$,*D*□□□□□***x***□□□□

□□□□□□□□□□□□□□,□□□□□□2□□,□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□2□□□□□□□

□□,□□□□□□□□□□□□□□□□□***x***□□□:

$$y=\operatorname{sign}(\boldsymbol{w}\boldsymbol{x}-b)$$

□□□sign□□□□□□,□□□□□□□□□□□□□□□,□□□+1;□□□□□,□□□-1□

SVM□□□□□□□□□□□□□□□□□□□**w**□*b*□□□□**w***□*b**□□□□□,□□□□□(model)*f*(**x**)□□□□□□:

$$f(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{w}^{*}\boldsymbol{x} - b^{*})$$

□□□,□□□,□**w***□□,□□□*b**□□□,sign□□□□□□□□□□□□(+1□□"□□□□□",-1□□"□□□□□□")□

□□,□□□□□□□□□**w***□*b**□?□□□□□□□□□□□□(optimization)□□,□□□□□□□□□

□□□□□□□□□?□□,□□□□□□□□□□□□□□□□□□1□□□□□□□□□□□□□□*i*=1,…,10 000□□□□□□□(**x***i*,*y**i*),□□**x***i*□□□□□□□*y**i*□□□+1□-1□□□□□□,□□□□□□□□□□□:

$$\boldsymbol{w}\boldsymbol{x}_i - b \geqslant +1, \quad 当\ y_i = +1\ 时$$
$$\boldsymbol{w}\boldsymbol{x}_i - b \leqslant -1, \quad 当\ y_i = -1\ 时$$

□□□□,□□□□□□□□□□□□□□□□□□□□□(margin)□□□(generalization),□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□**w**□□□□□□□□(Euclidean norm)$\|\boldsymbol{w}\|$,□□□□□□$\sqrt{\sum_{j=1}^{D} (w^{(j)})^2}$□□□

□□□□,□□□□□□□□□□□□□□□□□□□□□□□□:

$$\begin{aligned} &\text{minimize}\ \|\boldsymbol{w}\| \\ &\text{s. t. } y_i(\boldsymbol{w}\boldsymbol{x}_i - b) \geqslant 1, \quad 对于\ i = 1, \cdots, N \end{aligned}$$

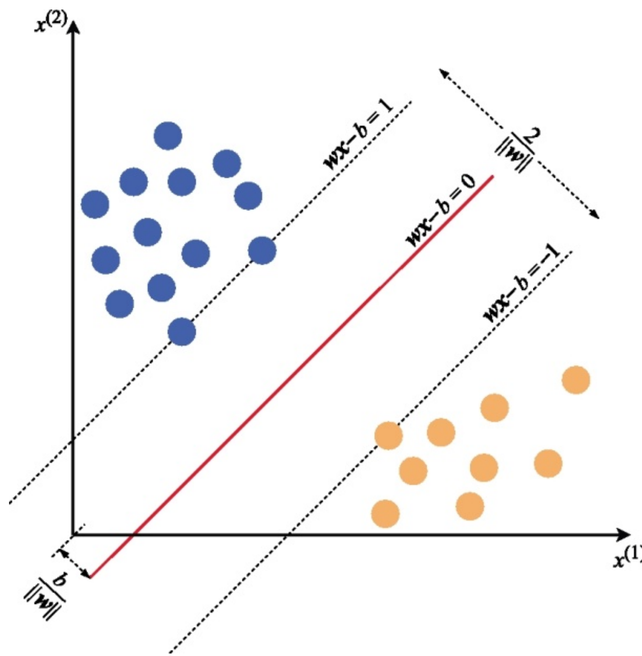
□□,□□□$y_i(\boldsymbol{wx}_i-b)\geq 1$□□□□□□□

□□□□□$\boldsymbol{w}^*$□b^*□□□□(statistical model),□□□□□□□□□□

□□□□(training)□

□1.1□□□□SVM□□□□□1.1□,□□□□□□□,□□□□□□□□

□□□;□□□□□□,□$\boldsymbol{wx}$-b=0□□□



□1.1 □□□□□□□□□□□SVM□□□□

□□,□□□$\boldsymbol{w}$□□□□□□□□□□□□□□□□□?□□1.1□□,$\boldsymbol{wx}$-b=1□$\boldsymbol{wx}$-b=-1□□□□□□□□□□□□□,□□□□□□$\frac{2}{\|\boldsymbol{w}\|}$□□□,□□$\|\boldsymbol{w}\|$□□,□□□□□□□

□□□□□□□□□□□□□□□□□□□□□(linear model)□□□□□□,□□□□□□□□□(□□□□□□□□□)□SVM□□□□□□□

(kernel)[illegible],[illegible](outlier)([illegible]),[illegible]SVM[illegible][5][illegible]3[illegible]SVM[illegible]

[illegible],[illegible],[illegible],[illegible](accuracy,[illegible])[illegible]

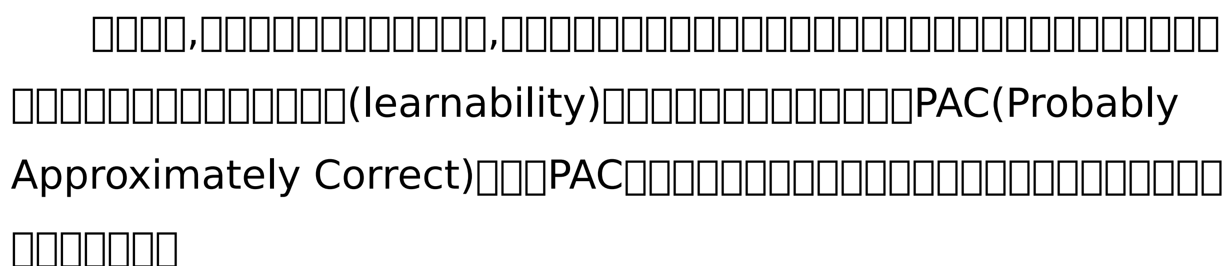
[illegible],[illegible]:[illegible],[illegible],[illegible]

1.4 [illegible]

[illegible]?[illegible]1.1[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible],SVM[illegible],[illegible]



□□□□,□□□□□□□□□□□□,□□□(learnability)□□□□□□□□□□□□PAC(Probably Approximately Correct)□□□PAC□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

---

[1] □□□□□□□□□□□□□□□□□□□□□

[2] □□□:“□□”□“□□”“□□”□□□□,□□□

[3] □□□□□□□□□□□□,□□0□256.34□1000□1000.2□□

[4] □□□:□□□□20000□□□□□□□□□□□

[5] □□□□□□□□□□□□□□□,□□(□□□□)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□

[illegible]2[illegible]　[illegible]

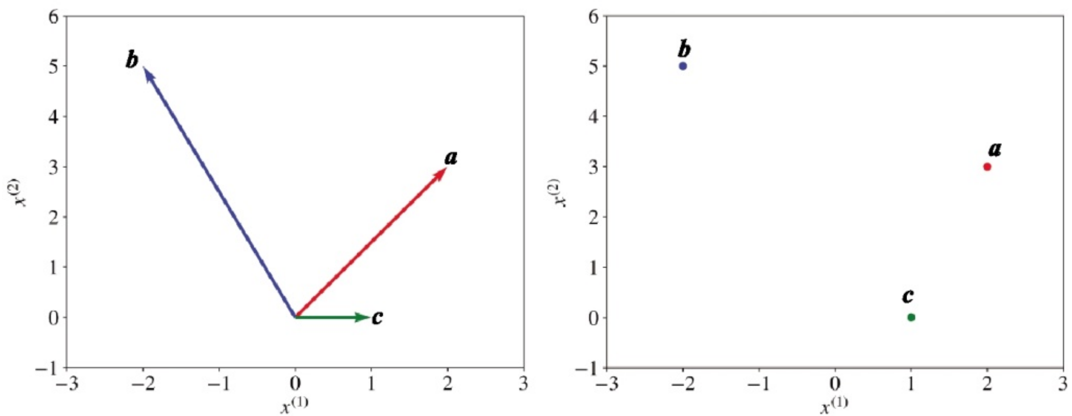
2.1　[illegible]

[illegible]

2.1.1　[illegible]

[illegible](scalar)[illegible],[illegible]15[illegible]-3.25[illegible],[illegible]x[illegible]a[illegible]

[illegible](vector)[illegible](ordered list)[illegible],[illegible],[illegible]$\boldsymbol{x}$[illegible]$\boldsymbol{w}$[illegible],[illegible],[illegible]2.1[illegible]3[illegible],[illegible]$\boldsymbol{a}$=[2,3][illegible]$\boldsymbol{b}$=[-2,5][illegible]$\boldsymbol{c}$=[1,0][illegible],[illegible]$w^{(i)}$[illegible]$x^{(j)}$[illegible],[illegible]i[illegible]j[illegible],[illegible]2.1[illegible]$\boldsymbol{a}$[illegible],$\boldsymbol{a}^{(1)}$[illegible]$\boldsymbol{a}^{(2)}$[illegible]2[illegible]3[illegible]



□2.1   3□□□□□□□□3□□□□3□□

□□□□□□,$x^{(j)}$□□□□□,□□x^2(□□)□x^3(□□)□□□□□□□□□□□□□□□□□□□□□□□□,□□□($x^{(j)}$)^2□□□□

□□□□□□□□□□□□□□,□□$x_i^{(j)}$□$x_{i,j}^{(k)}$□□□□□□□,□□□$x_{l,u}^{(j)}$□□□l□□□□□□u□□□□□j□□□□□□

□□□□(matrix)□□□□□□□□□□□□□□□□(array)□□□□□□□□□□□□□□□□□

$$\begin{bmatrix} 2 & 4 & -3 \\ 21 & -6 & -1 \end{bmatrix}$$

□□□□□□□□□□□□□,□□$\boldsymbol{A}$□$\boldsymbol{W}$□

□□□□(set)□□□□□□□□□□□□□□□(unordered collection)□□□□□□□□□□□□(□S)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□{1,3,18,23,235}□$\{x_1,x_2,x_3,x_4,\ldots,x_n\}$□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□a□b□□□□□□,□□a□b□□,□□□□□□□□□□□□□□[a,b];□□a□b□□□□□□,□□□□□□□□□□(a,b)□□□□□,□□[0,1]□□□□□□□□:0□0.000 1□0.25□0.784□0.999 5□1.0□□□,□□□□□$\mathbb{R}$□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□

$x \in S$□□“x□□□□□□S”□□□□□S_1□S_2□□□(intersection)□□□□□□□S_3,□□□$S_3 \leftarrow S_1 \cap S_2$□□□,{1,8}←{1,3,5,8}∩{1,8,4}□

S_1□S_2□□□(union)□□□□□□□□S_3,□□□$S_3 \leftarrow S_1 \cup S_2$□□□,{1,3,4,5,8}←{1,3,5,8}∪{1,8,4}□

2.1.2 □□□□□(Σ)□□

□□□□□$\boldsymbol{X}=\{x_1,x_2,\ldots,x_{n-1},x_n\}$,□□□□□$\boldsymbol{x}=[x^{(1)},x^{(2)},\ldots,x^{(m-1)},x^{(m)}]$□□□□□□□□□□,□□□Σ□□:

$$\sum_{i=1}^{n} x_i \stackrel{\text{def}}{=} x_1 + x_2 + \ldots + x_{n-1} + x_n$$

□□

$$\sum_{j=1}^{m} x^{(j)} \stackrel{\text{def}}{=} x^{(1)} + x^{(2)} + \ldots + x^{(m-1)} + x^{(m)}$$

□□,□□$\stackrel{\text{def}}{=}$□□□□“□□□”(is defined as)□

2.1.3 □□□(Π)□□

□□□□□(Π)□□□□□□□Σ□□□Π□□□□□□□□□□□□□□□□□:

$$\prod_{i=1}^{n} x_i \stackrel{\text{def}}{=} x_1 \cdot x_2 \cdot \ldots \cdot x_{n-1} \cdot x_n$$

□□,$a \cdot b$□□a□□b□□□□,□□□□□□□□□ab□□□a□□b□

2.1.4 □□□□

我们还可以使用类似的符号来表示集合运算:$S' \leftarrow \{x^2 | x \in S, x > 3\}$表示一个新集合的创建,即通过$S$中所有大于3的元素取平方后,放入新集合中得到$S'$。$|S|$表示的是集合$S$的元素数量。

2.1.5 向量运算

两向量相加:$\boldsymbol{x}+\boldsymbol{z}=[x^{(1)}+z^{(1)}, x^{(2)}+z^{(2)}, \ldots, x^{(m)}+z^{(m)}]$

两向量相减:$\boldsymbol{x}-\boldsymbol{z}=[x^{(1)}-z^{(1)}, x^{(2)}-z^{(2)}, \ldots, x^{(m)}-z^{(m)}]$

向量与标量相乘得到一个新的向量,例如:

$$\boldsymbol{x}c \stackrel{\text{def}}{=} \left[cx^{(1)}, cx^{(2)}, \ldots, cx^{(m)}\right]$$

两向量的点积(dot product)或内积(inner product)是一个标量,例如$\boldsymbol{w}\boldsymbol{x} \stackrel{\text{def}}{=} \sum_{i=1}^{m} w^{(i)} x^{(i)}$。一些书中使用$\boldsymbol{w} \cdot \boldsymbol{x}$来表示两个向量的点积,这两个向量必须有相同维度,否则点积就是未定义的。

一个矩阵$\boldsymbol{W}$和一个向量$\boldsymbol{x}$相乘得到另一个向量,例如下面的矩阵:

$$\boldsymbol{W} = \begin{bmatrix} w^{(1,1)} & w^{(1,2)} & w^{(1,3)} \\ w^{(2,1)} & w^{(2,2)} & w^{(2,3)} \end{bmatrix}$$

当向量参与矩阵运算的时候,在默认情况下,它就被表示为只有一列的矩阵。只有当向量和矩阵的行数相同时,矩阵才能和向量相乘。令向量为$\boldsymbol{x} \stackrel{\text{def}}{=} [x^{(1)}, x^{(2)}, x^{(3)}]$,则$\boldsymbol{W}\boldsymbol{x}$的计算结果是一个二维向量。计算过程如下:

$$
\boldsymbol{W}\boldsymbol{x} = \begin{bmatrix} w^{(1,1)} & w^{(1,2)} & w^{(1,3)} \\ w^{(2,1)} & w^{(2,2)} & w^{(2,3)} \end{bmatrix} \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{bmatrix}
\stackrel{\text{def}}{=} \begin{bmatrix} w^{(1,1)}x^{(1)} + w^{(1,2)}x^{(2)} + w^{(1,3)}x^{(3)} \\ w^{(2,1)}x^{(2)} + w^{(2,2)}x^{(2)} + w^{(2,3)}x^{(3)} \end{bmatrix}
= \begin{bmatrix} \boldsymbol{w}^{(1)}\boldsymbol{x} \\ \boldsymbol{w}^{(2)}\boldsymbol{x} \end{bmatrix}
$$

□□□□□□□□□□□□□□□□,□□***W***□□□,□***x***□□□□□□□□□□□□

□□□□□□,□□□□□□□□□□□□,□□□□□(transpose)□□□□□□□□□□□□□
$\boldsymbol{x}^{\mathrm{T}}$□□***x***□□□□□□□□:

$$
x = \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix}, \quad □ x^{\mathrm{T}} \stackrel{\text{def}}{=} \left[x^{(1)}, x^{(2)}\right]
$$

□□***x***□□□***W***□□□□□□□$\boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}$□□□□□□□:

$$
\boldsymbol{x}^{\mathrm{T}}\boldsymbol{W} = \left[x^{(1)}, x^{(2)}\right] \begin{bmatrix} w^{(1,1)} & w^{(1,2)} & w^{(1,3)} \\ w^{(2,1)} & w^{(2,2)} & w^{(2,3)} \end{bmatrix}
\stackrel{\text{def}}{=} \left[w^{(1,1)}x^{(1)} + w^{(2,1)}x^{(2)}, w^{(1,2)}x^{(1)} + w^{(2,2)}x^{(2)}, w^{(1,3)}x^{(1)} + w^{(2,3)}x^{(2)}\right]
$$

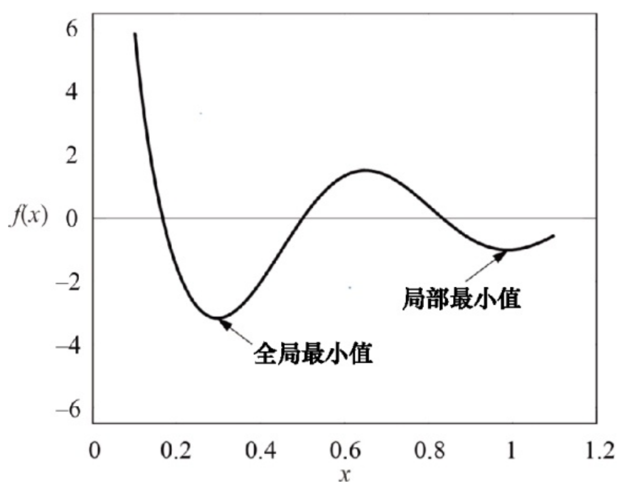
□□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□

2.1.6 □□

□□□□(function)□□□□□□□□*X*□□□□*x*,□□□□□□*Y*□□□□*y*□□□□□□□□□
X□□□□□□□□(domain),*Y*□□□(co-domain)□□□□□□□□□□□□□□□□,□□□
□□□□□□□*f*,□□□□□□□□*y*=*f*(*x*),□□“*y*□*x*□□□”□□□,*x*□□□□(argument),□
□□□;*y*□□□□,□□□□□

□□□□□□□□□□□□□□□(variable)□□□:x□□□f□□□□

□□□□□□$x=c$□□□□□□□□,□□x□□□$f(x) \geq f(c)$,□□□$x=c$□$f(x)$□□□□□□(local minimum)□□□□□(interval)□□□□□□□□□□□□□□□:□□□□□□□□□□□□□□□□□□□□□□□□□(open interval)□□□□□□□□□□□□□□,□□□□□□□□□:(0,1)□□□□□□□□□□□□0□□□1□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□2.2□□□



□2.2　□□□□□□□□□□

$\boldsymbol{y}=f(x)$□□□□□□□□□□□□□$\boldsymbol{y}$□□□□□□□□□□□□□□□□□□□□□

2.1.7　max□argmax

□□□□□□$A=\{a_1,a_2,\ldots,a_n\}$,□□□$\max\limits_{a \in A} f(a)$□□□□:□$A$□□□□□□□□,$f(a)$□□□□□□□□□$\operatorname{argmax}\limits_{a \in A} f(a)$□□□□$f(a)$□□□□□$a$□

□□□□□□□□□□□□□□□□□□,□□□□$\max\limits_{a} f(a)$□$\operatorname{argmax}\limits_{a} f(a)$□

□□□min□argmin□□□□max□argmax□□□

2.1.8 □□□□□

□□□$a \leftarrow f(x)$□□□□,□□a□□□□□□□$f(x)$□□□□a□□□□□□□□□□,$\boldsymbol{a} \leftarrow [a_1,a_2]$□□□□□□□□□□□□□□□□□□□$[a_1,a_2]$□

2.1.9 □□□□□

□□□□□□□(derivative)f'□□□□□□□□□□f□□(□□□)□□□□□□□□□□□□□□,□□5□-3,□□□□□f□□□□□□□□x□□□(□□□)□□□□□□□□□□□□f'□□□□□,f□□□□□□□□□□□□□□□□□□□□□□f'□□□x□□□□,f□□□□□;□□□□□□,□f□□□□□□□□□□□x□□0,□□□□□□□x□□□□(slope)□□□□□

□□□□□□□□□□□(differentiation)□

□□□□□□□□□□□□□□□□,□□$f(x)=x^2$,□$f'(x)=2x$;□□$f(x)=2x$,□$f'(x)=2$;□□$f(x)=2$,□$f'(x)=0$(□□□□□□$f(x)=c$□□□□□0)□

□□□□□□□□□□(chain rule)□□□□□□□□□□□□□□,□$F(x)=f(g(x))$,f□g□□□□,□$F'(x)=f'(g(x))g'(x)$□□□□□,□□$F(x)=(5x+1)^2$,□□$g(x)=5x+1$□$f(g(x))=(g(x))^2$□□□□□□□,□□□□$F'(x)=2(5x+1)g'(x)=2\times(5x+1)\times 5=50x+10$□

□□□□□□□□□□□□□□(□□□□□□□□□□,□□□□),□□□□□□□□□(generalization)□□□(gradient)□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□

□□,□□□□□□□$f([x^{(1)},x^{(2)}])=ax^{(1)}+bx^{(2)}+c$,$\frac{\partial f}{\partial x^{(1)}}$□□□$x^{(1)}$□□□$f$□□□□,□:

$$\frac{\partial f}{\partial x^{(1)}} = a + 0 + 0 = a$$

□□,a□□□$ax^{(1)}$□□□;□□0□□□$bx^{(2)}$□c□□□□□□□□□□$x^{(1)}$□□□□,□□$x^{(2)}$□□□□□□□,□□$bx^{(2)}$□□□□0□

□□,□□□$\frac{\partial f}{\partial x^{(2)}}$□□□$x^{(2)}$□□□$f$□□□□,□:

$$\frac{\partial f}{\partial x^{(2)}} = 0 + b + 0 = b$$

□□,□□f□□□,□□□∇f,□□□□□$\left[\frac{\partial f}{\partial x^{(1)}}, \frac{\partial f}{\partial x^{(2)}}\right]$□

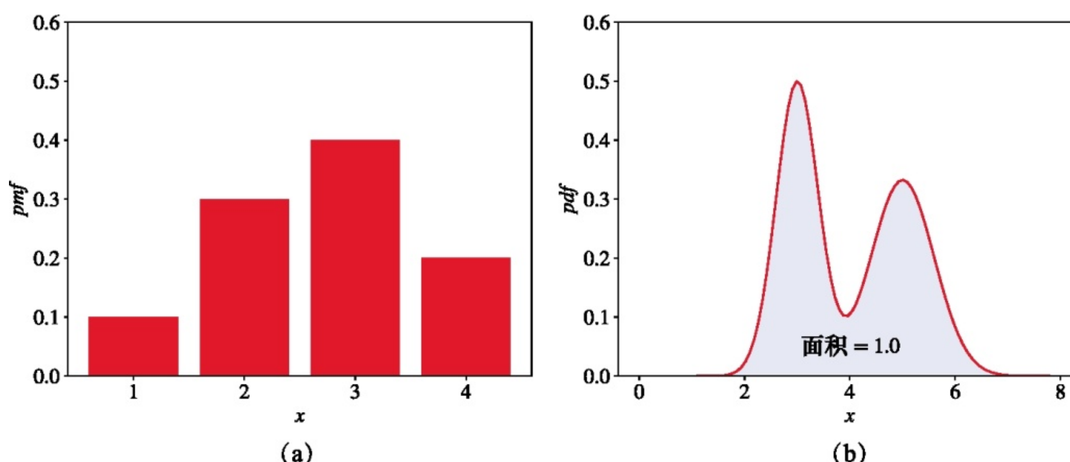
□□4□□,□□□□□□□□□□□□□□□□□□□□

2.2 □□□□

□□□□□□(random variable)□□□□□□□□(random phenomenon)□□□□□,□□□□□□□□□□□(□X)□□□□□□□□□□:□□□(discrete)□□□□(continuous)□

□□□□□□□□□(discrete random variable)□□□□□□□□□□□□□□□□□□□,□□,□□,□□□□1,2,3□□

□□□□□□□□□□□□□□□□□□(probability distribution),□□□□□□□□□□□□,□□□□□□□(probability mass function,pmf)□□□,Pr(X=□□)=0.3,Pr(X=□□)=0.45,Pr(X=□□)=0.25□□□□□□□□□□□□□□□□0,□□□□□□□□1[□□2.3(a)]□



□2.3　□□□□□□□□□□□□□□□□□

□□□□□□□(continuous random variable)□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□X□□□□□□□,□□□□□c□□Pr(X=c)□□0□□□□□□□□□(probability density function,pdf)□□□□□□□□□□□□□□□□□□,□□□□□□1[□□2.3(b)]□

□□□□□□□□□□X□k□□□,□□□$\{x_i\}_{i=1}^{k}$□□□□X□□(expectation)$\mathbb{E}[X]$□:

$$\mathbb{E}[X] \stackrel{\text{def}}{=} \sum_{i=1}^{k} \boldsymbol{x}_i \cdot \Pr(X=x_i) = x_1 \cdot \Pr(X=x_1) + x_2 \cdot \Pr(X=x_2) + \ldots + x_k \cdot \Pr(X=x_k) \tag{2.1}$$

□□,$\Pr(X=\boldsymbol{x}_i)$□□□□pmf,X□□x_i□□□□□□□□□□□□□□□□□□□□ (mean,average),□□□□□□□μ□□□□□□□□□□□□□□□□□□□(statistics)□ □□

□□□□□□□□□□□□□□□(standard deviation),□□□:

$$\sigma \overset{\text{def}}{=} \sqrt{\mathbb{E}\left[(X-\mu)^2\right]}$$

□□(variance),□□□σ^2□var(X),□□□:

$$\sigma^2 = \mathbb{E}\left[(X-\mu)^2\right]$$

□□□□□□□□□□□□□□□□□□□:

$$\sigma = \sqrt{\Pr(X=x_1)(x_1-\mu)^2 + \Pr(X=x_2)(x_2-\mu)^2 + \ldots + \Pr(X=x_k)(x_k-\mu)^2} \tag{2.2}$$

□□,$\mu=\mathbb{E}[X]$□

□□□□□□□□□□□□X□□□□:

$$\mathbb{E}[X] \overset{\text{def}}{=} \int_{\mathbb{R}} x f_X(x)\,\mathrm{d}x \tag{2.3}$$

□□,f_X□□□X□pdf,$\int_{\mathbb{R}}$□□□□xf_X□□□(integral)□□□

□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□1,□□□□□□□□□□$\int_{\mathbb{R}} f_X(x)\mathrm{d}x=1$□

□□□□□,f_X□□□□,□□□□□□□□□□□X□□□□□□□□□,□□□□□□□□□(example)□□□□□□□□□□□□(sample)□□□□(dataset)□

2.3 □□□□□

□□□□□□□,□□□□□□□□□□□□$S_X = \{x_i\}_{i=1}^N$□□□□□□f_X□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□(unbiased estimator)□

□□□□□□□□□□□□□□□□□□□S_X,□□□□□□□□□□□θ□□□□□□□□□□□,□□□$\hat{\theta}(S_X)$□□□□□□□□□

$$\mathbb{E}\left[\hat{\theta}\left(S_X\right)\right] = \theta$$

□□,$\hat{\theta}$□□□□□□□(sample statistic)□□□□□□□,□□$\hat{\theta}$□□□□□S_X□□,□□□□□□□□θ□□□X□□□□□□□□□□□□□□□□□X□□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□□□,□$\hat{\mu}$,□□X□□□μ□□□□□□□□□$\hat{\mu}$□□□□□□

□□□2.1□2.3□□□□□□□,□□□□□$\mathbb{E}[X]$□□□□□□$\frac{1}{N}\sum_{i=1}^{N} x_i$□□[□□□□□□□(sample mean)]□

2.4 □□□□□

□□□□(conditional probability)□□□□□□□□□Y□□y□□□□,□□□□□□□X□□□□□□□x□□□,□□□□Pr(X=x|Y=y)□□□□□□(Bayes'Rule)□□□□□□□□(Bayes'Theorem),□□:

$$\Pr(X=x|Y=y)=\frac{\Pr(Y=y|X=x)\Pr(X=x)}{\Pr(Y=y)}$$

2.5 □□□□

□□□□□□□□□□□□X□□,□□□□□□θ□□□□□□f_θ□□□,□□□□□□μ□σ□□□□□(Gaussian function)□□□

$$f_\theta(x)=\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

□□,$\theta\overset{\text{def}}{=}[\mu,\sigma]$□

□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□X□□□□□□□□□□□,□□□□□□□□□□□□□□θ□□□□:

$$\Pr(\theta=\hat{\theta}|X=x)\leftarrow\frac{\Pr(X=x|\theta=\hat{\theta})\Pr(\theta=\hat{\theta})}{\Pr(X=x)}$$

$$=\frac{\Pr(X=x|\theta=\hat{\theta})\Pr(\theta=\hat{\theta})}{\sum_{\tilde{\theta}}\Pr(X=x|\theta=\tilde{\theta})\Pr(\theta=\tilde{\theta})}$$

□□,$\Pr(X=x|\theta=\hat{\theta})\overset{\text{def}}{=}f_{\hat{\theta}}$□

□□□□□X□□□S,□θ□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□$\Pr(\theta=\hat{\theta})$□□□□□□□□□□□□$s\in S$□$\Pr(\theta=\hat{\theta})$□□□□□□□□$\sum_{\hat{\theta}}\Pr(\theta=\hat{\theta})=1$□□□□□□$\hat{\theta}$□□□□□□□□□□(prior)□

□□,□□□□□□□□$\hat{\theta}$□□□$\Pr(\theta=\hat{\theta}|X=x_1)$□□□□,□□□□□□2.4□$x=x_2\in S$□$\Pr(\theta=\hat{\theta}|X=x)$□□□□□□□□□,□□□□□,□□□□□□□□

$$\Pr(\theta=\hat{\theta}) \leftarrow \frac{1}{N}\sum_{x\in S}\Pr(\theta=\hat{\theta}|X=x)$$

□□□□□□□$\Pr(\theta=\hat{\theta})$□

□□□□□□,□□□□□□□□□□□(maximum likelihood)□□□□□□□□□□□θ^*:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{N} \Pr\left(\theta=\hat{\theta}\,|X| = x_i\right) \tag{2.5}$$

□□θ□□□□□□□□,□□□□□□□□□□□□□□[□□□□□(gradient descent)□]□□□□□□2.5□□□□□□□□□□□4□□□□□□□□□□□□,□□□□□□□2.5□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□[1]□

2.6 □□□□□

□□□□□□□□□□□□□□□□,□□□□□□□(□□□□)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□5□□□□□□□□□□□

□□□□□□□□□□,□□□

2.7 □□vs.□□

□□(classification)□□□□□□□□□□□□(unlabeled example)□□□□(label)□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible],[illegible](classification algorithm)[illegible](labeled example)[illegible],[illegible](model)[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible](classe)[illegible](“[illegible]”/“[illegible]”,“[illegible]”/“[illegible]”),[illegible](binary classification)[[illegible](binomial)][illegible]3[illegible][2],[illegible](multiclass classification)[illegible]

[illegible],[illegible],[illegible]7[illegible]

[illegible](regression)[illegible],[illegible]([illegible])[illegible],[illegible]

[illegible](regression learning algorithm)[illegible],[illegible]

2.8 [illegible]vs.[illegible]

[illegible],[illegible]1[illegible]SVM[illegible]SVM[illegible],[illegible] $\boldsymbol{w}^*$ [illegible] b^* [illegible]

[illegible],[illegible]**k**[illegible](k-Nearest Neighbor,kNN)[illegible],kNN[illegible] k [illegible]“[illegible]”[illegible],kNN[illegible]

2.9 □□□□vs.□□□□

□□□□(shallow learning)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(neural network)□□□□□,□□□□□□□□□□□□□□□□(layer)□□□□□,□□□□□□□□□□□(deep neural networks)□□□□□□□□□(□□□“□□□□”)□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□

□□□□□6□□□□□□□□□□□□□

---

[1] □□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□

[2] □□□□□□□□□□□□

[illegible]3[illegible] [illegible]

[illegible],[illegible]5[illegible],[illegible]

3.1 [illegible]

[illegible](linear regression)[illegible],[illegible](linear combination)[illegible]

3.1.1 [illegible]

[illegible]$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$,$N$[illegible],$\boldsymbol{x}_i$[illegible]$i=1,\ldots,N$[illegible]$D$[illegible]$y_i$[illegible][1],[illegible]$x_i^{(j)}$,$j=1,\ldots,D$[illegible]

[illegible]x[illegible]$f_{\boldsymbol{w},b}(\boldsymbol{x})$:

$$f_{\boldsymbol{w},b}(x) = \boldsymbol{w}\boldsymbol{x} + b \tag{3.1}$$

$f_{\boldsymbol{w},b}$[illegible]f[illegible],[illegible]$\boldsymbol{w}$[illegible]b[illegible],$\boldsymbol{w}$[illegible]D[illegible],b[illegible]

[illegible],[illegible]y[illegible]$\boldsymbol{x}$[illegible],[illegible]$y \leftarrow f_{\boldsymbol{w},b}(\boldsymbol{x})$[illegible],[illegible]$(\boldsymbol{w}, b)$[illegible]$(\boldsymbol{w}^*, b^*)$[illegible]

[illegible],[illegible]

[illegible]3.1[illegible]SVM[illegible],[illegible]sign[illegible],[illegible],[illegible]SVM[illegible],[illegible],[illegible]

[illegible],[illegible]3.1[illegible]([illegible])[illegible]([illegible])[illegible],[illegible]x_{new}[illegible]y_{new}[illegible]D[illegible](D>1),[illegible]([illegible])[illegible](D>2)[illegible]

[illegible]3.1 10[illegible]

[illegible]:[illegible]3.1[illegible],[illegible]y_{new}[illegible]

3.1.2 [illegible]

[illegible],[illegible]$\boldsymbol{w}^*$[illegible]b^*:

$$\frac{1}{N}\sum_{i=1\ldots N}\left(f_{w,b}\left(\boldsymbol{x}_i\right)-y_i\right)^2 \tag{3.2}$$

□□□□,□□□□□□□□□□□□□□□□□□□□(objective function),□□□□“□□”□□□3.2□,$(f_{\boldsymbol{w},b}(\boldsymbol{x}_i)-y_i)^2$□□□□□(loss function),□□□□□□□□□□□□□□□□□□,□3.2□□□□□□□□□□(squared error loss)□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□[□□□□□□(cost function)]□□□□□□□□□□□□□□□□□□□,□□□□□□□(empirical risk)□□□□□□□□□(□□□□)□□□□□□□□□□□□□□□□□□□□□□□

□□,□□□□□□□□□□□□□□□□(quadratic function)□?□□□□□□□□y_i□□□□$f(\boldsymbol{x}_i)$□□□□□□□□□□□?□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□□□□□,□□,□□y_i□$f(\boldsymbol{x}_i)$□□□□□□□□□□□□□,□□□□□□(binary loss)(y_i□$f(\boldsymbol{x}_i)$□□□□□□□1,□□□0)□□□□□□?

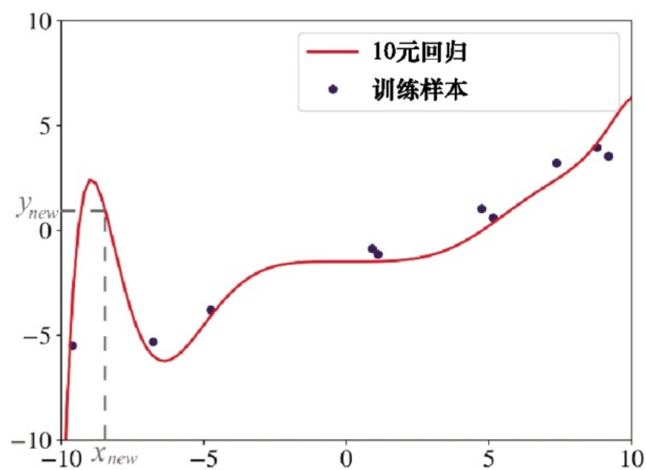
□□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□?□□,□□□□□□□□□,□□□□□□□□□□

□□,□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□□□,□□□□□□□□□□
- □□□□□□□□□□□,□□□□□□□□□□

[illegible],[illegible]?[illegible],[illegible](overfitting)[illegible],[illegible]

[illegible]3.2[illegible]([illegible])[illegible]3.1[illegible],[illegible]([illegible])[illegible]10[illegible](polynomial regression)[illegible],[illegible]5[illegible]



[illegible]3.2 [illegible]

[illegible],[illegible]?[illegible]?1805[illegible],[illegible]·[illegible]·[illegible](Adrien-Marie Legendre)[illegible]?[illegible],[illegible],[illegible],[illegible],[illegible](closed form solution)[illegible][[illegible](gradient descent)],[illegible],[illegible]

如图所示,红色直线一侧的点为一类,另一侧的点为另一类。很明显3号和4号被分错,但是除此之外,其他点都分对了。

那么,如何从训练样本中获得这条直线呢?我们将在3.2节中,通过后面介绍的逻辑回归,将上式的右侧改为0[2]的同时,得到了最优的参数,即分别为w^*和b^*。

3.2 逻辑回归模型

逻辑回归模型(logistic regression)[3]是一种常见的分类器,用于对样本属于某一类别的概率进行预测,同时也可以判断样本的类别是哪一类。

本节将介绍逻辑回归模型的基本原理、训练方法以及怎样使用训练后的模型进行分类预测。

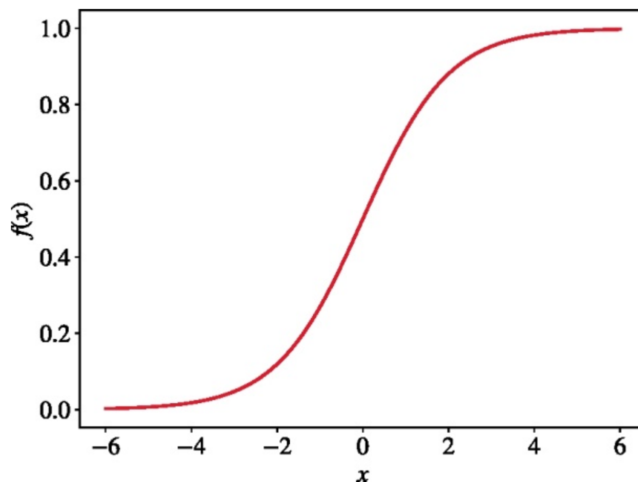
3.2.1 模型原理

逻辑回归的输入中,对于每个样本x_i都有一个标签y_i。在二分类中,y_i的取值为,对应样本所属的类别。我们希望通过线性函数wx_i+b的值来预测样本属于某一类的概率,而y_i取值仅为离散值。

为了解决这个问题,需要将线性函数的值进行转换,将其映射到一个取值范围有限的区间内,例如使其最小值接近0而最大值接近1,即将线性函数的值映射到(0,1)区间上。我们需要一个函数,使得对于x的值较大或0时,对应于x的较小的值;否则,x的值较大。这个函数就是标准逻辑函数(standard logistic function),也称为sigmoid函数,它的数学表达式具体如下所示:

$$f(x)=\frac{1}{1+\mathrm{e}^{-x}}$$

□□,e□□□□□□□□□□,□□□□□□(Euler's Number)□□□□□□□□□exp(x)□□ e^x□sigmoid□□□□□□□3.3□□□



□3.3 □□□□□□□

□□□□□□□□□□□□□:

$$f_{\boldsymbol{w},b}\left(\boldsymbol{x}\right) \overset{\text{def}}{=} \frac{1}{1+\mathrm{e}^{-(\boldsymbol{w}\boldsymbol{x}+b)}} \tag{3.3}$$

□□□□□□□□□□□□□□□$\boldsymbol{w}\boldsymbol{x}+b$□

□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□:□□□□□□□□□□□$\boldsymbol{w}$ □b□,□□□□$f(\boldsymbol{x})$□□□□□□y_i□□□□□□□□□□□,□□□□□□□□□□□□□0.5,□□□□$\boldsymbol{x}$ □□□□□;□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□5□□□□□□□□ □□□□□□□□□

□□□,□□□□□□□□□□□$\boldsymbol{w}^*$□b^*□?□□□□,□□□□□□,□□□□□□□□□□□,□□□ □□□□(Mean Squared Error,MSE)□

3.2.2 □□□□

□□□□□□□□,□□□□□□□□□□□□□□□(likelihood)□□□□□□□,□□□□□□□
□□□□□□□□□□□□□□□□□□□□

□□□□,□□□□□□□□□□□□□□($\boldsymbol{x}_i$,y_i)□□□,□□□□□□□□(□□□)□□□□□□□$\hat{\boldsymbol{w}}$
□$\hat{b}$□□□,□□□3.3□□□□□$\boldsymbol{x}_i$□□□$f_{\hat{\boldsymbol{w}},\hat{b}}$□□□,□□□□□□□$0<p<1$□□□□□$y_i$□□□,□
□□□□□□,y_i□□□□□□□□p□□□□,□□y_i□□□,□y_i□□□□□□□□$1-p$□

□□□□□□□□□(optimization criterion)□□□□□□(maximum likelihood)□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□:

$$L_{\boldsymbol{w},b} \stackrel{\text{def}}{=} \prod_{i=1\ldots N} f_{\boldsymbol{w},b}(\boldsymbol{x}_i)^{y_i}\left(1-f_{\boldsymbol{w},b}(\boldsymbol{x}_i)\right)^{(1-y_i)} \tag{3.4}$$

□□□$f_{\boldsymbol{w},b}(\boldsymbol{x}_i)^{y_i}(1-f_{\boldsymbol{w},b}(\boldsymbol{x}_i))^{(1-y_i)}$□□□□,□□□□□□□□□:□$y_1=1$□□
$f_{\boldsymbol{w},b}(\boldsymbol{x}_i)$□□,□□□$(1-f_{\boldsymbol{w},b}(\boldsymbol{x}_i))$□□□□□□□□,□□$y_1=1$,□$(1-y_1)=0$□□□□□□□
□□0□□□□1,□□$(1-f_{\boldsymbol{w},b}(\boldsymbol{x}_i))^{(1-y_i)}$□□1□□□,□□$y_1=0$,$f_{\boldsymbol{w},b}(\boldsymbol{x}_i)^{y_i}$□□1□

□□□□□□□□,□□□□□□□□□□□□□□Π□□□□□□□□□□□□□Σ□□□□□,□□□
N□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□□□□□□,□□□□□□)□□□□□□
□□□□□(log-likelihood)□:

$$\log L_{\boldsymbol{w},b} \stackrel{\text{def}}{=} \ln\left(L_{\boldsymbol{w},b}(\boldsymbol{x})\right) = \sum_{i=1}^{N} y_i \ln f_{\boldsymbol{w},b}(\boldsymbol{x}) + (1-y_i)\ln\left(1-f_{\boldsymbol{w},b}(\boldsymbol{x})\right)$$

□□ln□□□□□□□□□(strictly increasing function),□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□,□□□□□(gradient descent)□□□□□□,□□□□□□□□□

3.3 □□□□□

□□□(decision tree)□□□□□□□□□□□□□(graph)□□□□□□□□□,□□□□i□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□;□□,□□□□□□□□□□□□□□□□□□□□□(leaf node)□,□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□,□□□□□□□□□□□□□□□□

3.3.1 □□□□

□□□□□□□□:□□□□□{0,1}□□□□□□□□□□□□,□□□□□□□□□□□□□□□

3.3.2 □□□□

□□□□□□□□□□□□□□□□,□□□□□□□□□□ID3□□□□

□□□□□□□□□□□□□□□:

$$\frac{1}{N}\sum_{i=1}^{N} y_i \ln f_{\mathrm{ID3}}\left(\boldsymbol{x}_i\right)+\left(1-y_i\right)\ln\left(1-f_{\mathrm{ID3}}\left(\boldsymbol{x}_i\right)\right)$$

(3.5)

□□,f_{ID3}□□□□□□□

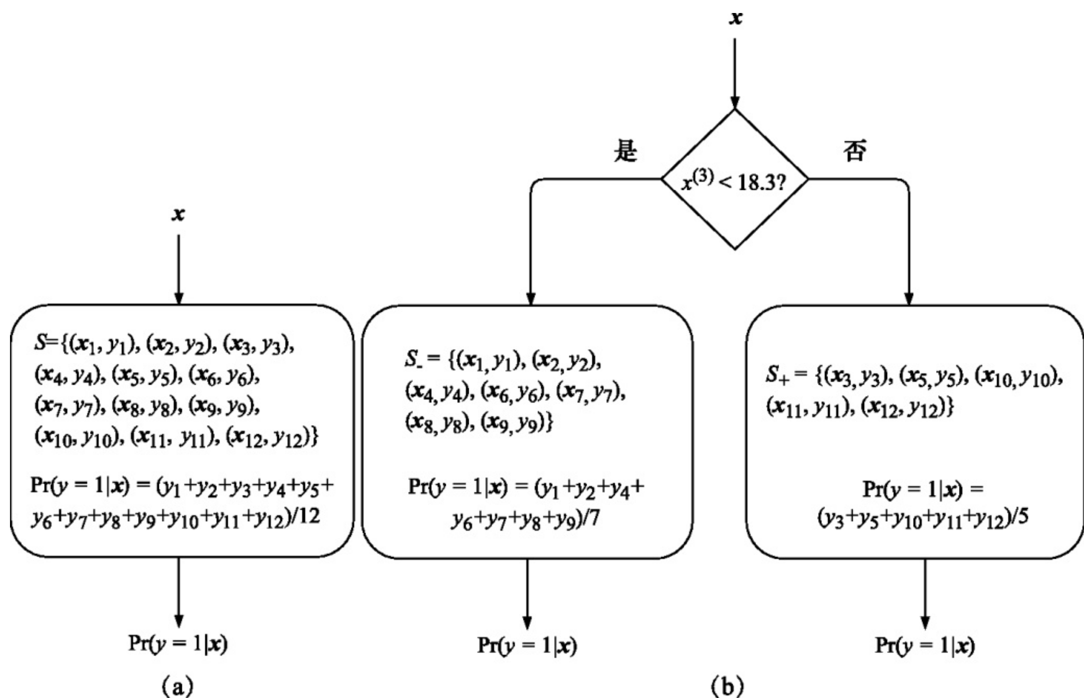
□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□(parametric model),□f_{w^*,b^*};□ID3□□□□□□□□□□□□□□□□□□□(nonparametric model),□$f_{\mathrm{ID3}}(\boldsymbol{x}) \stackrel{\mathrm{def}}{=} \Pr(y=1|\boldsymbol{x})$□

ID3□□□□□□□□□□□S□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□,$S \stackrel{\mathrm{def}}{=} \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$□□□□□□□□□□□:

$$f_{\mathrm{ID3}}^S \stackrel{\mathrm{def}}{=} \frac{1}{|S|} \sum_{(\boldsymbol{x},y)\in S} y$$

(3.6)

□□□□□□□□□$\boldsymbol{x}$□□□,$f_{\mathrm{ID3}}^S(\boldsymbol{x})$,□□□□□□□□3.4(a)□,□□□□□□□□□12□□□□□□□□



□3.4 □□□□□□□□□□□□

□:□□□□S□□12□□□□□□(a)□□,□□□□□□□□□□□,□□□□□□□□□□□□(b)□□□□□□□□□□□□□□□□,□□□□3□□□□18.3,□□□□□□□□□□□□□□□□□□□□

□□,□□□□□□□□j=1,...,D□□□□□t,□□S□□□□□□□: $S_- \stackrel{\text{def}}{=} \{(\boldsymbol{x}, y) | (\boldsymbol{x}, y) \in S, x^{(j)} < t\}$□$S_+ \stackrel{\text{def}}{=} \{(\boldsymbol{x}, y) | (\boldsymbol{x}, y) \in S, x^{(j)} \geqslant t\}$□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□(j,t)□S_+□S_-□□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□S_+□S_-(□□□□□□□□□□□□□□□□□□□□□□)□□3.4(b)□□□□□□□3.4(a)□□□□□□□□□□□□□□

□□,□□□□□□□□□□□□□?□ID3□,□□□□(entropy)□□,□□□□□□□□□□□□S□□□□□□:

$$H(S) \stackrel{\text{def}}{=} -f_{\text{ID3}}^{S} \ln f_{\text{ID3}}^{S} - \left(1 - f_{\text{ID3}}^{S}\right) \ln \left(1 - f_{\text{ID3}}^{S}\right)$$

□□□□□□□□□j□□□□□t□□□□□□□□□,□□□□□$H(S_{-},S_{+})$□□□□□□□□□□□□□

$$H\left(S_{-}, S_{+}\right) \stackrel{\text{def}}{=} \frac{\left|S_{-}\right|}{|S|} H\left(S_{-}\right)+\frac{\left|S_{+}\right|}{|S|} H\left(S_{+}\right) \tag{3.7}$$

□ID3□□□□□□,□□□□□□□,□□□□□3.7□□□,□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□,□□□□□□□□□□:

- □□□□□□□□□□□□□□(□3.6)□□□□□
- □□□□□□□□□□□□□
- □□□□□□□□□□□ε□(□□□□□□□□[4])□
- □□□□□□□□□□□(□□□□□□□)□

□□□□□□□,□□□□□□□□□□□□□□(□□□□□□□□□),ID3□□□□□□□□□□□□□□□□□□□□□□(backtracking)□□□□□□□□,□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□**C4.5**□□□ID3,□□□□□□:

- □□□□□□□□□□□□□
- □□□□□□□□□□
- □□□□□□□□(bottom-up)□“□□”(pruning)□□□□□□□□□□

□□□□□□□□:□□□□□□□□,□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□

[illegible]:[illegible]*S*[illegible],[illegible]0;[illegible],[illegible]1,[illegible],[illegible]

3.4 [illegible]

[illegible]1[illegible],[illegible]:

1[illegible],[illegible],[illegible]?

2[illegible],[illegible](higher-order polynomial)[illegible],[illegible]?

[illegible]3.5[illegible],[illegible]([illegible]),[illegible]3.5(b)[illegible],[illegible],[illegible]

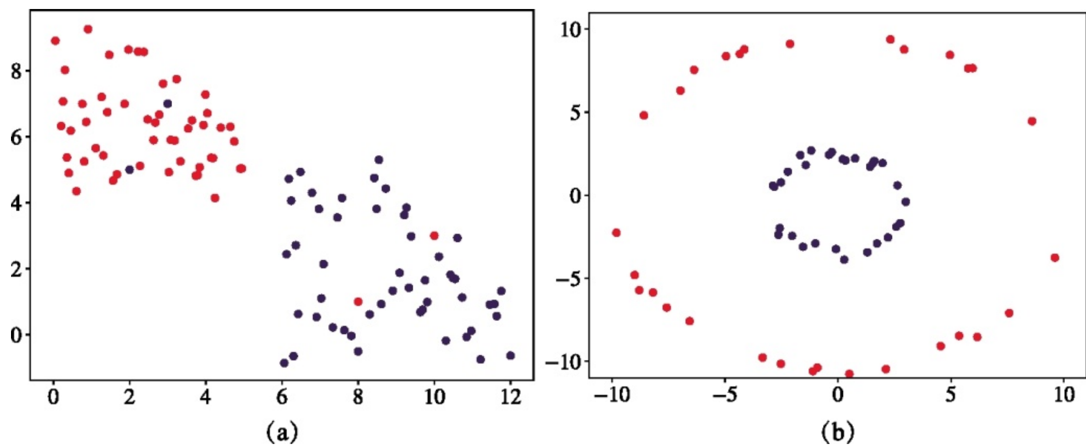


图3.5　线性可分和线性不可分

对于线性SVM,数据点需要满足以下约束条件:

$$\boldsymbol{w}\boldsymbol{x}_i - b \geqslant +1, \quad \text{当 } y_i = +1 \text{ 时}$$
$$\boldsymbol{w}\boldsymbol{x}_i - b \leqslant -1, \quad \text{当 } y_i = -1 \text{ 时}$$

(3.8)

此外,为了最大化间隔,需要最小化分类器的参数,在实际优化过程中,为了数学上的便利,通常将$\|\boldsymbol{w}\|$的最小化替换为$\frac{1}{2}\|\boldsymbol{w}\|^2$的最小化。这样处理后,优化问题转化成了一个典型的二次规划(quadratic programming)问题。综上所述,SVM的优化问题可表述为:

$$\min\frac{1}{2}\|\boldsymbol{w}\|^2, \quad \text{满足},y_i\left(\boldsymbol{x}_i\boldsymbol{w}-b\right)-1\geqslant 0, i=1,\ldots,N$$

(3.9)

3.4.1　核心技巧

[illegible]SVM[illegible],[illegible](hinge loss)[illegible]

[illegible]:max(0,1-y_i($\boldsymbol{wx}_i$-b))[illegible]

[illegible]3.8[illegible],[illegible]0[illegible],[illegible]$\boldsymbol{wx}_i$[illegible],[illegible]0[illegible],[illegible]

[illegible],[illegible]:

$$C\|\boldsymbol{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N}\max\left(0, 1 - y_i\left(\boldsymbol{wx}_i - b\right)\right)$$

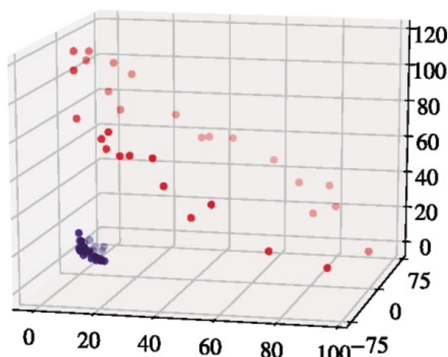
[illegible],[illegible]$\boldsymbol{x}_i$[illegible]C[illegible]ID3[illegible]ϵ[illegible]d,C[illegible]SVM[illegible](soft-margin)SVM[illegible]

[illegible],[illegible]C[illegible],[illegible]SVM[illegible],[illegible]C[illegible],SVM[illegible],[illegible],C[illegible]([illegible])[illegible]([illegible])[illegible]

3.4.2 [illegible]

SVM[illegible],[illegible],[illegible]SVM[illegible],[illegible](implicitly)[illegible](kernel trick)[illegible]

□□□□□□□□□□3.6□□,□□□□□□□□□□(mapping)ϕ:$\boldsymbol{x}$a$\phi(\boldsymbol{x})$□□□□□□□□□□□□□□□□□□□□□□□□□□□□,$\phi(\boldsymbol{x})$□□□□$\boldsymbol{x}$□□□□□□□□□3.5(□)□□□□□□□□,□□□□□□□$\boldsymbol{x}$=[q,p]□□□□□□□□□(□□3.6)□,$\phi([q,p]) \stackrel{\text{def}}{=} (q^2, \sqrt{2}qp, p^2)$□□□,$\cdot^2$□□□□□□□□□,□□□□□□□□□□□□□□□□□□□



□3.6　□□□□□□□□□□□□,□3.5(b)□□□□□□□□□□□□

□□,□□□□□□□□□□□□□ϕ□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□SVM□□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□

□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□[kernel function,□□□□(kernel)]□□□□□□□□□□,□□□□□□□□SVM□□□□□□□□□□$\boldsymbol{w}$□b□□□□□

□□□□□3.9□□□□□□□□□□□□□(method of Lagrange multiplier)□□□□□□□3.9□□,□□□□□□□□□□□□□□:

$$\max_{\alpha_1 \ldots \alpha_N} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} y_i \alpha_i (\boldsymbol{x}_i \boldsymbol{x}_k) y_k \alpha_k$$

□□□$\sum_{i=1}^{N} \alpha_i y_i = 0$□□$\alpha_i \geq 0, i=1,\ldots,N$

□□,α□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□(convex quadratic optimization problem),□□□□□□□□□□□□□□□

□□□□□□□□□□□□,□□□□□□□□□□□□,□$\boldsymbol{x}_i\boldsymbol{x}_k$□□□□□□□□□□□□□□□□□□□□□□,□□□$\boldsymbol{x}_i$□$\boldsymbol{x}_k$□□□□$\phi(\boldsymbol{x}_i)$□$\phi(\boldsymbol{x}_k)$□□□□□□□□□□□□□□□

□□,□□□□□$\boldsymbol{x}_i\boldsymbol{x}_k$□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□(q_1,p_1)□(q_2,p_2)□□□($q_1^2,\sqrt{2}q_1p_1,p_1^2$)□($q_2^2,\sqrt{2}q_2p_2,p_2^2$)□□□□□□□($q_1^2q_2^2+2q_1q_2p_1p_2+p_1^2p_2^2$),□□□□□□$(q_1,p_1)$□$(q_2,p_2)$□□□$(q_1q_2,p_1p_2)$,□□□□□□□□□□□□□($q_1^2q_2^2+2q_1q_2p_1p_2+p_1^2p_2^2$)□

□□□□□□□□□□□□□,□□□□$k(\boldsymbol{x}_i,\boldsymbol{x}_k)\stackrel{\text{def}}{=}(\boldsymbol{x}_i\boldsymbol{x}_k)^2$□□□□□□□,□□□□□□□□□□(Radial Basis Function,RBF)□:

$$k\left(\boldsymbol{x},\boldsymbol{x}'\right)=\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\sigma^2}\right)$$

□□,$\|\boldsymbol{x}-\boldsymbol{x}'\|^2$□□□□□(Euclideandistance)□□□□□□□□□□□□□□:

$$\begin{aligned} d\left(\boldsymbol{x}_i,\boldsymbol{x}_k\right) &\stackrel{\text{def}}{=} \sqrt{\left(x_i^{(1)}-x_k^{(1)}\right)^2+\left(x_i^{(2)}-x_k^{(2)}\right)^2+\ldots+\left(x_i^{(D)}-x_k^{(D)}\right)^2} \\ &= \sqrt{\sum_{j=1}^{D}\left(x_i^{(j)}-x_k^{(j)}\right)^2} \end{aligned}$$

RBF[illegible]σ,[illegible]

3.5 k[illegible]

k[illegible](k-Nearest Neighbor,kNN)[illegible],[illegible]kNN[illegible]$\boldsymbol{x}$[illegible],kNN[illegible]$\boldsymbol{x}$[illegible]k[illegible],[illegible],[illegible]

[illegible],[illegible](consine similarity)[illegible]:

$$s\left(\boldsymbol{x}_i, \boldsymbol{x}_k\right) \stackrel{\text{def}}{=} \cos\left(\angle\left(\boldsymbol{x}_i, \boldsymbol{x}_k\right)\right) = \frac{\sum_{j=1}^{D} x_i^{(j)} x_k^{(j)}}{\sqrt{\sum_{j=1}^{D}\left(x_i^{(j)}\right)^2}\sqrt{\sum_{j=1}^{D}\left(x_k^{(j)}\right)^2}}$$

[illegible]0[illegible],[illegible],[illegible]1[illegible],[illegible]0[illegible],[illegible]-1[illegible],[illegible]-1[illegible](Chebychev distance)[illegible](Mahalanobis distance)[illegible](Hamming distance)[illegible]k[illegible],[illegible]([illegible])[illegible]10[illegible]

[1] [illegible]$y_i \in \mathbb{R}$[illegible]y_i[illegible],$\mathbb{R}$[illegible],[illegible]

[2] [illegible],[illegible]0[illegible],[illegible]0[illegible],[illegible]

[3] [illegible]:[illegible],[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[4] [illegible]5[illegible],[illegible]

[illegible]4[illegible] [illegible]

4.1 [illegible]

[illegible],[illegible]3[illegible]:

- [illegible]
- [illegible]([illegible])[illegible]
- [illegible]

[illegible],[illegible]([illegible],[illegible],[illegible]),[illegible]([illegible]k[illegible])[illegible]k[illegible],[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible],[illegible]“[illegible]”[illegible]“[illegible]”[illegible],[illegible]

[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible],[illegible]SVM[illegible],[illegible],[illegible]SVM,[illegible],[illegible],[illegible],[illegible],[illegible]

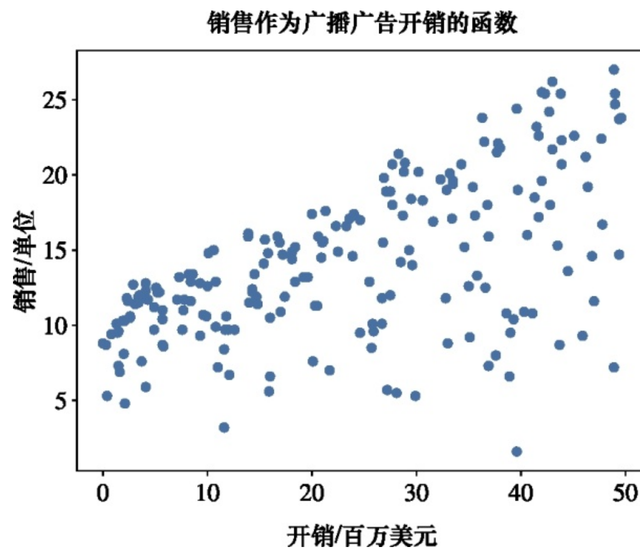
[illegible]

4.2 [illegible]

[illegible],[illegible][1][illegible]Python[illegible],[illegible],[illegible],[illegible]:w[illegible]b[illegible]:[illegible]$w^{(1)}$,$w^{(2)}$[illegible]b,[illegible]$w^{(1)}$,$w^{(2)}$,$w^{(3)}$[illegible]b,[illegible]

[illegible],[illegible]([illegible])[illegible]:[illegible],[illegible],[illegible]

[illegible]200[illegible],[illegible]200[illegible],[illegible](x_i,y_i)=([illegible],[illegible])[illegible]4.1[illegible]



[illegible]4.1 [illegible]

□:y□□□□□□□(□□□□□□□□□),x□□□□□□□□□,□□□□□□□□□□(□□□□)□

□□	□□/□□□□	□□/□□
1	37.8	22.1
2	39.3	10.4
3	45.9	9.3
4	41.3	18.5
…	…	…

□□□□□□,□□□□□□□□□□:$f(x)=\boldsymbol{wx}+b$□□□□□□w□b□□□□,□□□□□□
□□□□□□□,□□□□□□□□□□□□w□b□:

$$l \stackrel{\text{def}}{=} \frac{1}{N}\sum_{i=1}^{N}\left(y_i-(wx_i+b)\right)^2$$

□□□□□□□□□□□□□□□□□□:

$$\frac{\partial l}{\partial w}=\frac{1}{N}\sum_{i=1}^{N}-2x_i\left(y_i-(wx_i+b)\right);$$
$$\frac{\partial l}{\partial b}=\frac{1}{N}\sum_{i=1}^{N}-2\left(y_i-(wx_i+b)\right)$$

(4.1)

□□$(y_i-(wx+b))^2$□w□□□□,□□□□□□□□□□□□□,$f=f_2(f_1)$□□□□□□□□,$f_1=y_i-(wx+b)$,□$f_2 = f_1^2$□□□f□w□□□□,□□□□f□f_2□□□□,□□$2(y_i-(wx+b))$(□□□□□,□□□□□□$\frac{\partial f}{\partial x}x^2 = 2x$)□□□,□□$y_i-(wx+b)$□$w$□□□□$-x$□□□□□□,□$\frac{\partial l}{\partial w} = \frac{1}{N}\sum_{i=1}^{N} -2x_i(y_i - (wx_i + b))$□□□□□,□$l$□$b$□□□□$\frac{\partial l}{\partial b}$□

□□□□□□□(epoch)□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□[2]□□w←0□b←0□□□□□$\frac{\partial l}{\partial w}$□$\frac{\partial l}{\partial b}$□□□□$\frac{-2}{N}\sum_{i=1}^{N} x_i y_i$□$\frac{-2}{N}\sum_{i=1}^{N} y_i$□□□□□□,□□□□□□□□$w$□$b$□□□□□□□□□□□□$\alpha$□□□

$$w \leftarrow w - \alpha\frac{\partial l}{\partial w};$$
$$b \leftarrow b - \alpha\frac{\partial l}{\partial b}$$

(4.2)

□□□□□□□□(□□□□)□□□,□□□□□□□□□□□□□□□□□□□□[3]□□,□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□,□□□□□□□□(□□□□□)□□□□□□□(□□□□□),□□□□□□□,□□□□□□□□□

□□□□□,□□□4.1□□w□b□□□□□□□;□□□□□,□□□□□□□□□,□□□□□□□□□□□□w□b□□□□□□□□□□□□□,□□□□□□□□

□□□□□□□□□□Python□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□Python□□□□□□□□□□

□□□□□,□□□w□b□□□□□□□

```python
def update_w_and_b(spendings, sales, w, b, alpha):
    dl_dw = 0.0
    dl_db = 0.0
    N = len(spendings)

    for i in range(N):
        dl_dw += -2*spendings[i]*(sales[i] - (w*spendings[i] + b))
        dl_db += -2*(sales[i] - (w*spendings[i] + b))

    # update w and b

    w = w - (1/float(N))*dl_dw*alpha
    b = b - (1/float(N))*dl_db*alpha

    return w, b
```

□□□for□□□□□□□□□□□□□□□□

```python
def train(spendings, sales, w, b, alpha, epochs):
    for e in range(epochs):
        w, b = update_w_and_b(spendings, sales, w, b, alpha)

        # log the progress
        if e % 400 == 0:
            print("epoch:", e, "loss: ", avg_loss(spendings,
                sales, w, b))
    return w, b
```

□□train□□□□avg_loss□□□□□□□□□□□□,□□□□□□:

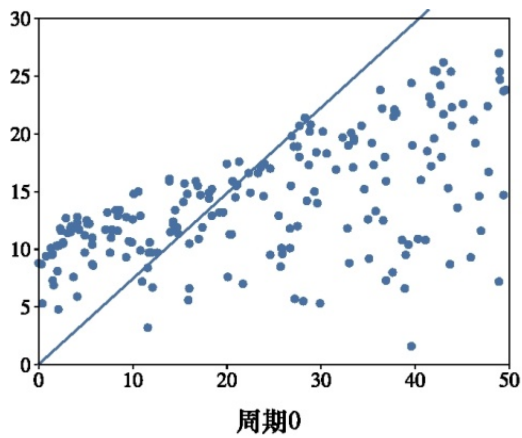
```python
def avg_loss(spendings, sales, w, b):
    N = len(spendings)
    total_error = 0.0
    for i in range(N):
        total_error += (sales[i] - (w*spendings[i] + b))**2
    return total_error / float(N)
```

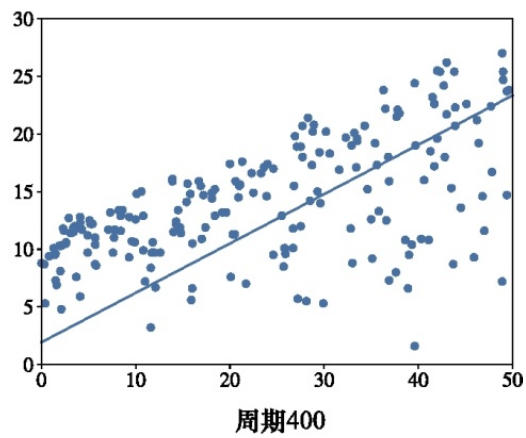
□□□□□□α=0.001□w=0.0□b=0.0,□□□□1 500□□train□□,□

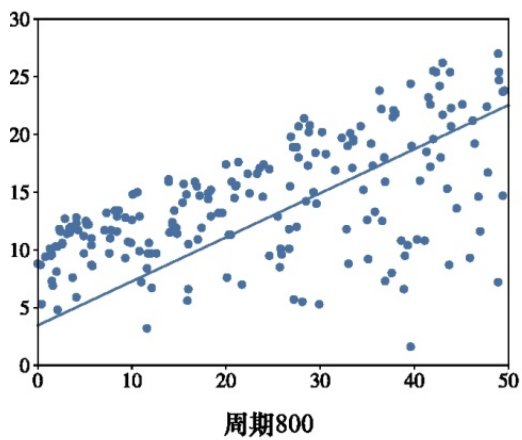
□□□□□□(□□□□□)□

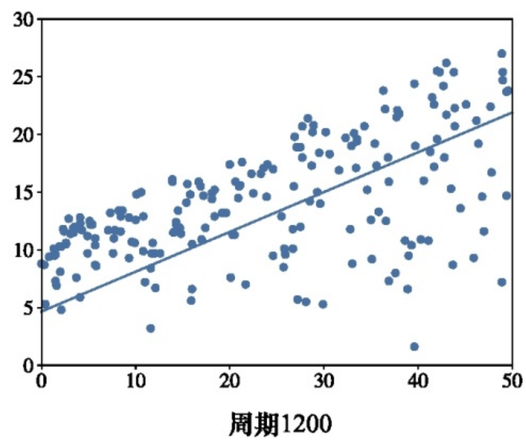
```
epoch:  0 loss: 92.32078294903626
epoch:  400 loss: 33.79131790081576
epoch:  800 loss: 27.9918542960729
epoch:  1200 loss: 24.33481690722147
epoch:  1600 loss: 22.028754937538633
...
epoch:  2800 loss: 19.07940244306619
```

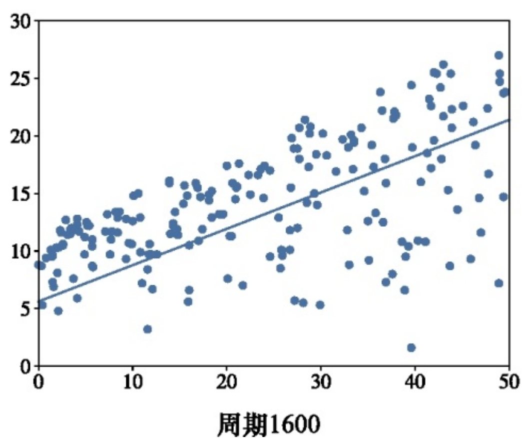
[illegible],[illegible]train[illegible],[illegible]4.2[illegible]

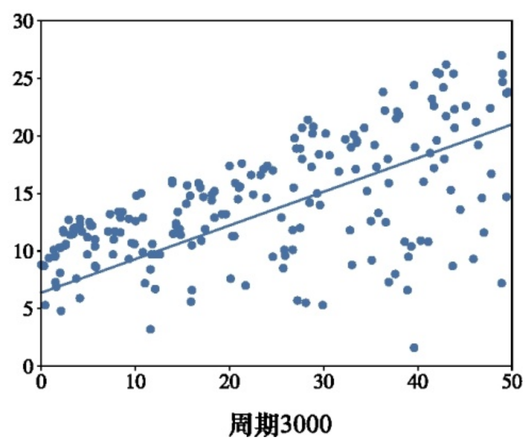












[illegible]4.2 [illegible]

[illegible],[illegible]w[illegible]b[illegible],[illegible]:

```
31  def predict(x, w, b):
32      return w*x + b
```

□□□□□□□□□

```
33  w, b = train(x, y, 0.0, 0.0, 0.001, 15000)
34  x_new = 23.0
35  y_new = predict(x_new, w, b)
36  print(y_new)
```

□□□□□□□13.97□

□□□□□□□□□α□□□□□□□□□,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□(minibatch Stochastic Gradient Descent,minibatch SGD)□□□□□□□□□□□□,□□□□□□□□□□□□(□□)□□□□,□□□□□□□□□SGD□□□□□“□□”□□□,Adagrad□□□□□□□□□,□□□□□□□□□□□□□□□□□□α:□□□□□□□α□□,□□□□□□□(Momentum)□□□□SGD□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□SGD□□□,□RMSprop□Adam□

□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□(□□□□□□□□□□□□)□

4.3 □□□□□□□□□□□□

□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□(library),□□□□□□□□□□□□(open sourced)□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□scikit-learn,□Python□C□□□□□□□□□scikit-learn□□□□□□:

```python
def train(x, y):
    from sklearn.linear_model import LinearRegression
    model = LinearRegression().fit(x,y)
    return model

model = train(x,y)

x_new = 23.0
y_new = model.predict(x_new)
print(y_new)
```

[illegible]13.97[illegible]?[illegible]LinearRegression([illegible])[illegible],[illegible],[illegible]LogisticRegression([illegible])[illegible]SVC([illegible]scikit-learn[illegible]),DecisionTreeClassifier([illegible])[illegible]NearestNeighbor(k[illegible])[illegible]scikit-learn[illegible]

4.4 [illegible]

[illegible],[illegible]([illegible],SVM[illegible]C,ID3[illegible]ϵ[illegible]d);[illegible]([illegible])[illegible],[illegible]α[illegible]

[illegible],[illegible],[illegible],[illegible]"[illegible]"[illegible],[illegible]"[illegible]""[illegible]""[illegible]",[illegible]SVM[illegible]k[illegible]([illegible])[illegible],scikit-learn[illegible]

[illegible],[illegible]SVM,[illegible],[illegible]([illegible])[illegible],[illegible],[illegible]

[illegible],[illegible]SVM[illegible]k[illegible],[illegible],[illegible],[illegible]0[illegible]1[illegible],[illegible]

[illegible]([illegible]SVM)[illegible],[illegible]([illegible]SGD[illegible]/SGD[illegible] scikit-learn[illegible] PassiveAggressiveClassifier/PassiveAggressiveRegressor)[illegible],[illegible]

[illegible],[illegible]([illegible]SVM[illegible]k[illegible])[illegible];[illegible],[illegible]

[illegible],[illegible]

---

[1] [illegible],[illegible],[illegible],[illegible]

[2] [illegible]([illegible])[illegible],[illegible]([illegible]0[illegible]0[illegible]),[illegible]

[3] [illegible]

□5□ □□□□

□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□□□□
□□scikit-learn□model=LogisticRegression().fit(x,y)□□□□□□□
□□

5.1 □□□□

□□□□□,□□□□□:“□□□□5□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□”□□,□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□
(dataset)□

□□□□□□□,□1□□□□,□□□□□□□□□□(labeled example)□□,□□□
$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$□□□□□$N$□□□□□$\boldsymbol{x}_i$□□□□□□□(feature vector)□□□□□□□□
□j=1,…,D□□□□□□□□□□□□□□□□□□□□□□(feature),□□□$x^{(j)}$□

□□□□(feature engineering)□□□□□□□□□□□□□□□□□□□□□□□□,
□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□

□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□(session)□□□□(□□□□□),□□□□□

(request)之间的响应时间(response time)、页面会话的长度或指定位置中的温度等特征。对用户行为有价值的信息量较大(informative)的特征:如在线收集的数据显示活跃的用户会比不活跃的用户更易下单。在点击中,有更大预测能力(predictive power)的特征显而易见,例如会话中(session)的点击次数,但生成这样的合适特征仍有一定的难度和挑战性。

在特征工程中,我们有诸多方法应对处理数据中不同类型的特征,以下将介绍几种在实践中较为常见的特征处理方法。

5.1.1 独热编码

在处理数据的过程中,许多机器学习的算法很难直接处理分类型的特征,比如"国籍"与"被访问的页面",这时我们通常需要对分类特征进行编码。

下面我们以一个"颜色"为例,它有3个可能的值"红""黄""绿",将此特征的可能值编码为具有3个维度的向量:

$$
\begin{aligned}
\text{红} &= [1,0,0] \\
\text{黄} &= [0,1,0] \\
\text{绿} &= [0,0,1]
\end{aligned}
\tag{5.1}
$$

可以看出,某些学习模型使用向量计算来对数据进行处理,如果我们单纯地将颜色转化为数字1、数字2或者数字3来代表红、黄、绿三种颜色,由于不同的数值在模型中代表其有顺序之分,比如回归算法会误以为数据中存在先后顺序[1]这种关系,所以需要引入独热编码来解决此类问题,从而不会混淆变量。

5.1.2 分箱

[illegible](binning)[illegible](bucketing)[illegible],[illegible],[illegible],[illegible]:[illegible]0~5[illegible],6~10[illegible],11~15[illegible],[illegible]

[illegible],[illegible]*j*=4[illegible],[illegible]3[illegible],“[illegible]1”“[illegible]2”“[illegible]3”,[illegible]*j*=123[illegible]*j*=124[illegible]*j*=125[illegible],[illegible]*x*[illegible]$x_i^{(4)}=7$,[illegible]$x_i^{(124)}=1$;[illegible]$x_i^{(4)}=13$,[illegible]$x_i^{(125)}=1$,[illegible]

[illegible],[illegible]“[illegible]”,[illegible],[illegible]

5.1.3 [illegible]

[illegible](normalization)[illegible],[illegible][-1,+1][illegible][0,1][illegible]

[illegible],[illegible]350[illegible]1450,[illegible]350,[illegible]110,[illegible][0,1][illegible]

[illegible]:

$$\bar{x}^{(j)} = \frac{x^{(j)} - \min^{(j)}}{\max^{(j)} - \min^{(j)}}$$

[illegible],$\min^{(j)}$[illegible]$\max^{(j)}$[illegible]*j*[illegible]

那么这是为什么呢?因为,在最小化损失函数的梯度下降过程中,当特征的取值在不同的数量级时(见第4章中,梯度下降的计算方式),梯度下降会按比例同时更新各个特征的权重$w^{(1)}$和$w^{(2)}$等,这样就会导致取值较大的$w^{(1)}$和$w^{(2)}$更新较慢。比如$x^{(1)}$的范围是[0,1 000],而$x^{(2)}$的范围则是[0,0.000 1],那么较小的特征取值在更新时会很难收敛。

所以,数据的标准化会使梯度下降的收敛更快,而使数据的分布变为一个均值为零、方差为一的分布(正态分布)。

5.1.4 标准化

标准化(standardization)又叫**z**值标准化(z-score normalization),是将数据的特征值缩放为满足μ=0和σ=1标准正态分布的一种方法,μ为均值(在全部样本中某个特征的均值),σ为该特征的方差。

标准化的数据的计算(即z值计算)方式如下所示:

$$\hat{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}}$$

看了前面的数据归一化,标准化和归一化的原理一样吗?为什么需要标准化而不是归一化呢?其实,标准化和归一化都是为了统一量纲,避免某一维特征过大,而导致学习时,计算出的结果偏向于这个特征,从而导致模型对结果的把握度不准确。

标准化和归一化有些许的区别,具体区别如下:

- 从原理上,归一化是将数据缩放到同一个尺度上面。
- 而标准化是将特征数据转化成与数据(所有数据均值)相关的,并服从正态分布。
- 归一化对于特征中最大或最小的值(异常值),有较高的敏感程度;而标准化对于数据的处理中,由于是以一种“平均”的方式进行缩放。

- [illegible],[illegible]

[illegible],[illegible]

5.1.5 [illegible]

[illegible],[illegible],[illegible],[illegible]

[illegible]:

- [illegible]([illegible],[illegible])[illegible]
- [illegible]([illegible])[illegible]
- [illegible]

5.1.6 [illegible]

[illegible],[illegible]:

$$\hat{x}^{(j)} \leftarrow \frac{1}{M}\sum_{i=1}^{N} x_i^{(j)}$$

[illegible]$M<N$,M[illegible]j[illegible],[illegible]j[illegible]

[illegible],[illegible][0,1],[illegible]2[illegible]-1[illegible],[illegible],[illegible],[illegible]

和序数型特征(即有序离散特征),可把其编码到区间[-1,1],并用独热码编码标称特征。0值的缺失特征转换为数值特征的第一种方法。

还可以使用回归方法,即缺失值预测。对于具有缺失特征的样本,构建新数据集。其中$[x_i^{(1)},x_i^{(2)},\ldots,x_i^{(j-1)},x_i^{(j+1)},\ldots,x_i^{(D)}]$称为新的特征$\hat{\boldsymbol{x}}_i$,令$\hat{y}_i \leftarrow x^{(j)}$为标签,$j$为缺失特征的位置,该数据集由所有特征$\hat{\boldsymbol{x}}_i$和标$\hat{y}_i$的样本组成的集合,就可以构建回归模型来预测缺失特征j的值,模型的输入是$(\hat{\boldsymbol{x}},\hat{y})$。

此外,当数据集很大,并且缺失值数量有限时,可以直接用模型预测的值进行填充,或者可以使用原特征和新增的维度来判断。假设特征向量维度为D的数据集中,特征$j=12$存在缺失值。对于每一个样本x来,可以添加一个特征$j=D+1$。若特征12存在缺失值,则新特征为1;否则新特征为0。对于缺失值,可以用特征值填充为0或其他任意数值。

实际应用,可尝试多种填充技术,然后选择效果最好的模型来进行填充和训练。

如果训练数据有缺失值,那么测试数据中出现缺失值的时候,也必须使用同样的技术来填补缺失值。

5.2 学习算法选择

通常情况下,选择什么样的机器学习算法取决于具体的业务场景,可以从以下几个方面进行考虑,具体来说,在选择之前,可以先思考下列问题:建模的目的是什么,数据集有什么特点等等,对这些问题进行思考,有助于更快地选择合适的学习算法,提高建模的效率。

- 可解释性

模型是否需要解释,也就是说我们是否需要知道模型是如何做出预测的?一些非常精确的学习算法通常被称为"黑箱",比如神经网络或集成模型,它们所做的预测能够很准确,但很难理解它们的预测过程,也就是很难解释,但可解释性对于

[illegible]

[illegible],[illegible]k[illegible],[illegible]

- [illegible]vs.[illegible]

[illegible]?[illegible],[illegible];[illegible],[illegible](incremental learning algorithm),[illegible]

- [illegible]

[illegible]?[illegible]?[illegible],[illegible]([illegible]);[illegible],[illegible]SVM[illegible]

- [illegible]vs.[illegible]

[illegible],[illegible]?[illegible],[illegible],[illegible]

- [illegible]

[illegible]?[illegible]?[illegible],[illegible]SVM[illegible];[illegible],[illegible]6[illegible]7[illegible]

- [illegible]

□□□□□□□□□□□□□□□□□□□□□?□□□□□□□□□□□□□□□□□□□□□,□□□□
□□□□□□□□□□,□□
□□□□□□□(□□□□□)□□□□□□□□□(CPU)□□□□□□□□□□□□□□□□□

- □□□□

□□□□□□□□□□□□□□□□□□?□□□□□□□□□□□□□□□□□□□□?□□SVM□□
□□□□□□□□□□(□□)□□□□□□,□□□□□□□□□□□,□□k□□□□□□□□□□□□□□□□
□□□□,□□□□□[2]□

□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□,□□□□scikit-learn□,□□□□□□□□□□□□□□,□□5.1□□□

□5.1 scikit-learn□□□□□□□□

5.3　3个数据集

说到这里，除了前面说的“训练集”和“测试集”之外，我们还需要有一个数据集。在工作中，我们通常会把所有数据分为如下3个数据集：

- 训练数据
- 验证数据
- 测试数据

当然，对小数据集，可以分为训练集和测试集，但是对大数据量的建模，最好分成3个集合：训练数据用来拟合模型参数，验证数据用于模型选择和调参，最后的测试数据用来评估模型的性能。其中，评估模型性能时使用的数据集，在模型训练之前就已经分隔出来，这个集合也叫作保持集(holdout set)。

3个数据集如何分配，在数据量的规模上没有统一的规定，通常的做法是，训练集占总数据量的70%，验证集占15%，测试集也占15%。在数据量大时，比如海量数据，验证集和测试集的数据量相对可以更少一些，比如训练集占到95%，验证集和测试集合，各自占总数据的2.5%。

那么，为什么要分成3个数据集，特别是验证集的作用是什么？答案就是：在模型的选择中，如果只用训练集和测试集，多次在训练集上拟合模型，并在测试集上验证、调参，这样无意中就会让模型“看到”测试集的数据，这就存在信息泄露的风险，导致最后的评估结果过于乐观，模型在测试集上表现不错，但在新数据上表现不好，失去了模型评估的客观性。

那么，应该如何避免模型的过拟合问题呢？这就是接下来要介绍的模型评估方法，这些方法可以用于模型的选择，也能帮助我们选择更好的模型。

5.4　模型的评估方法

[illegible],[illegible](bias)[illegible],[illegible],[illegible](high bias),[illegible](underfit)[illegible],[illegible]:

- [illegible]([illegible],[illegible])[illegible]
- [illegible]

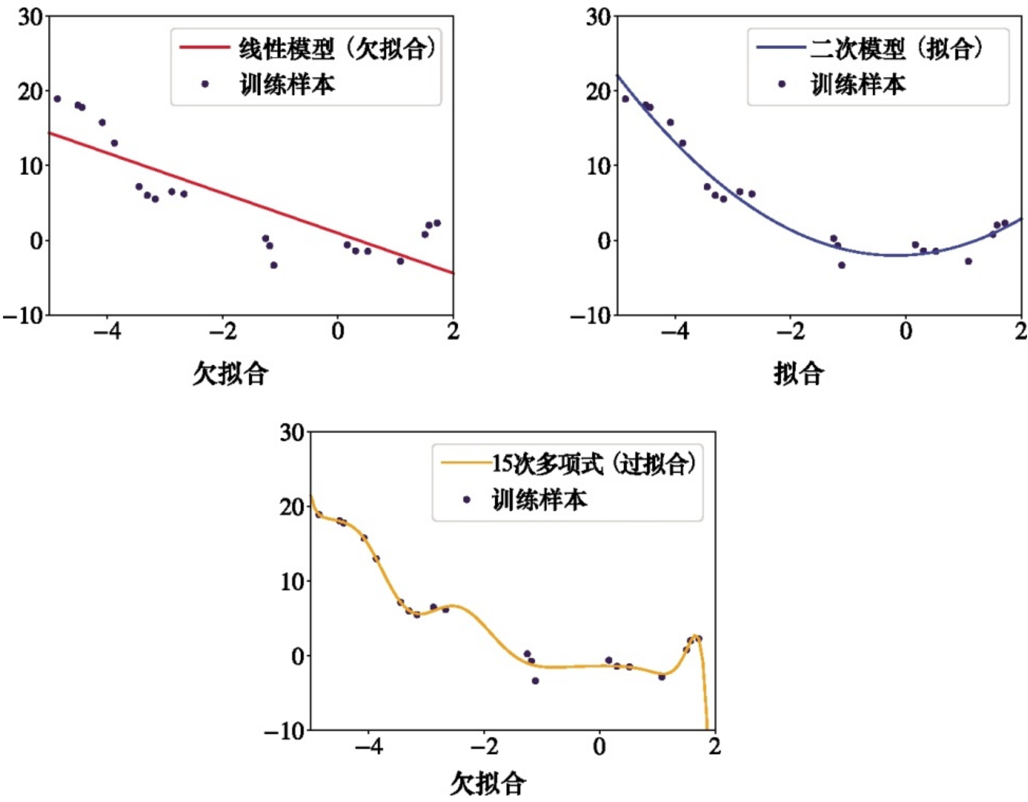
[illegible]:[illegible],[illegible]:[illegible],[illegible],[illegible]3[illegible],[illegible]

[illegible],[illegible]

[illegible](overfitting)[illegible],[illegible]3[illegible],[illegible]:

- [illegible]([illegible],[illegible])[illegible]
- [illegible],[illegible]

[illegible],[illegible]:[illegible](high variance)[illegible],[illegible],[illegible],[illegible]:[illegible]



30
20
10
0
−10
−4
−2
0
2
线性模型（欠拟合）
训练样本
欠拟合
二次模型（拟合）
训练样本
拟合
15次多项式（过拟合）
训练样本
欠拟合

(1)□□□□□□□□□□(□□□□□□□□□□□□□□,□□□□□□□□□□□SVM□□RBF□,□□□□□□□□/□□□□□□□□□□□)□

(2)□□□□□□□□□□□□(□□,□□□□□□□□9□□□□□□□□□□□□□□□)□

(3)□□□□□□□,□□□□□□□□□

(4)□□□□□□□□□

□□□(regularization)□□□□□□□□□□□□□□□

5.5 □□□

□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□-□□□□(bias-variance tradeoff)□

L1□L2□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□:

$$\min_{\boldsymbol{w},b} \frac{1}{N} \sum_{i=1}^{N} \left(f_{\boldsymbol{w},b}\left(\boldsymbol{x}_i\right) - y_i\right)^2 \tag{5.2}$$

□□L1□□□□□□□□□□□:

$$\min_{\boldsymbol{w},b}\left[C\left|\boldsymbol{w}\right|+\frac{1}{N}\sum_{i=1}^{N}\left(f_{\boldsymbol{w},b}\left(\boldsymbol{x}_i\right)-y_i\right)^2\right]$$

(5.3)

□□, $\left|\boldsymbol{w}\right|\stackrel{\text{def}}{=}\sum_{j=1}^{D}\left|w^{(j)}\right|$,C□□□□□□□□□□□□□□□□□□C□□0,□□□□□□□□□□□□□□□□□□□□,□□□C□□□□□,□□□□□□□□□□□□□□□□□□$w^{(j)}$□□□□□□0□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□C□□,□□□□□□□□□□,□□□□□□□□□□□□□□□□□6□□,□□□□□□□□□□

□□L2□□□□□□□□□:

$$\min_{\boldsymbol{w},b}\left[C\|\boldsymbol{w}\|^2+\frac{1}{N}\sum_{i=1}^{N}\left(f_{\boldsymbol{w},b}\left(\boldsymbol{x}_i\right)-y_i\right)^2\right],$$

□ $\|\boldsymbol{w}\|^2\stackrel{\text{def}}{=}\sum_{j=1}^{D}(w^{(j)})^2$ □

(5.4)

□□□□□□,L1□□□□□□□□□□□(sparse model)□□□□□C□□□□□□,□□□□□□□□(□□□□□□□□,□□□$w^{(j)}$)□□□0□□□,L1□□,□□□□□□□□□□□□□□□□□,□□L2□□□□□□□□□□□,L2□□□□□□□,□□□□□□□□□□□□□□

□□□□□□□□□□(elastic net regularization)□□□□□L1□L2□□,L1□L2□□□□□□□□□□□□□□□□□□L2□□□□□□(ridge

regularization),[illegible]L1[illegible](lasso)[illegible]

[illegible],L1[illegible]L2[illegible],[illegible]

[illegible],[illegible](dropout)[illegible](batch normalization)[illegible],[illegible]:[illegible](data augmentation)[illegible](early stopping)[illegible]8[illegible]

5.6 [illegible]

[illegible],[illegible]?[illegible]

[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible]:[illegible],[illegible](mean model)[illegible],[illegible]

[illegible],[illegible][3](MSE)[illegible]MSE[illegible],[illegible],[illegible]“[illegible]”[illegible],[illegible]/[illegible]

[illegible],[illegible]:

- 混淆矩阵。
- 精确率/召回率。
- 调和平均
- 精确-召回率曲线
- ROC和曲线下面积

在讨论指标前,首先要介绍混淆矩阵的概念,这是理解这些性能指标的关键。

5.6.1 混淆矩阵

混淆矩阵(confusion matrix)是一个矩阵,用于展示学习算法的性能。它是一种特定的表格布局,可以直观显示监督学习算法的性能,其中列代表预测类别的实例,行代表实际类别的实例。以下展示了一个二分类问题的混淆矩阵,可以区分“实际恶性”和“实际非恶性”。

	实际恶性(预测)	实际非恶性(预测)
实际恶性(实际)	23(真阳,TP)	1(假阴,FN)
实际非恶性(实际)	12(假阳,FP)	556(真阴,TN)

在以上混淆矩阵中,在24个实际恶性肿瘤中,算法正确预测了其中的23个为恶性。在这种情况下,我们称这23个为真阳(true positive,TP),即TP=23。算法错误地将一个恶性肿瘤预测为良性,这就是1个假阴性(false negative,FN),即FN=1。同时,在568个实际良性肿瘤中,556个被正确预测,这556个是真阴(true negative,TN),即TN=556。同样,12个良性肿瘤被错误预测为恶性,这12个是假阳(false positive,FP),即FP=12。

(precision) (recall)

5.6.2

$$查准率 \stackrel{\text{def}}{=} \frac{真正}{真正 + 假正}$$

$$查准率 \stackrel{\text{def}}{=} \frac{真正}{真正 + 假正}$$

- [illegible](SVM[illegible])[illegible]
- [illegible],[illegible]
- [illegible],[illegible],[illegible]([illegible]),[illegible]0.9[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]

5.6.3 [illegible]

[illegible](accuracy)[illegible],[illegible]:

$$\text{准确率} \stackrel{\text{def}}{=} \frac{\text{真正} + \text{真负}}{\text{真正} + \text{真负} + \text{假正} + \text{假负}} \tag{5.5}$$

[illegible],[illegible]/[illegible],[illegible]:[illegible],[illegible],[illegible]:[illegible],[illegible]

5.6.4 [illegible]

[illegible],[illegible](cost-sensitive accuracy)[illegible],[illegible]([illegible]):[illegible],[illegible]

(TP)□□□(TN)□□□□□□□□□□□□□□□□□□□□□□□□,□□□5.5□□□□□□

5.6.5 ROC□□□□□

ROC□□(□□□“□□□□□□□□□”,□□□□□□□□)□□□□□□□□□□□□□□□□
ROC□□□□□□□(□□□□□□□□)□□□□(□□□□□□□□□),□□□□□□□□□□□□□□

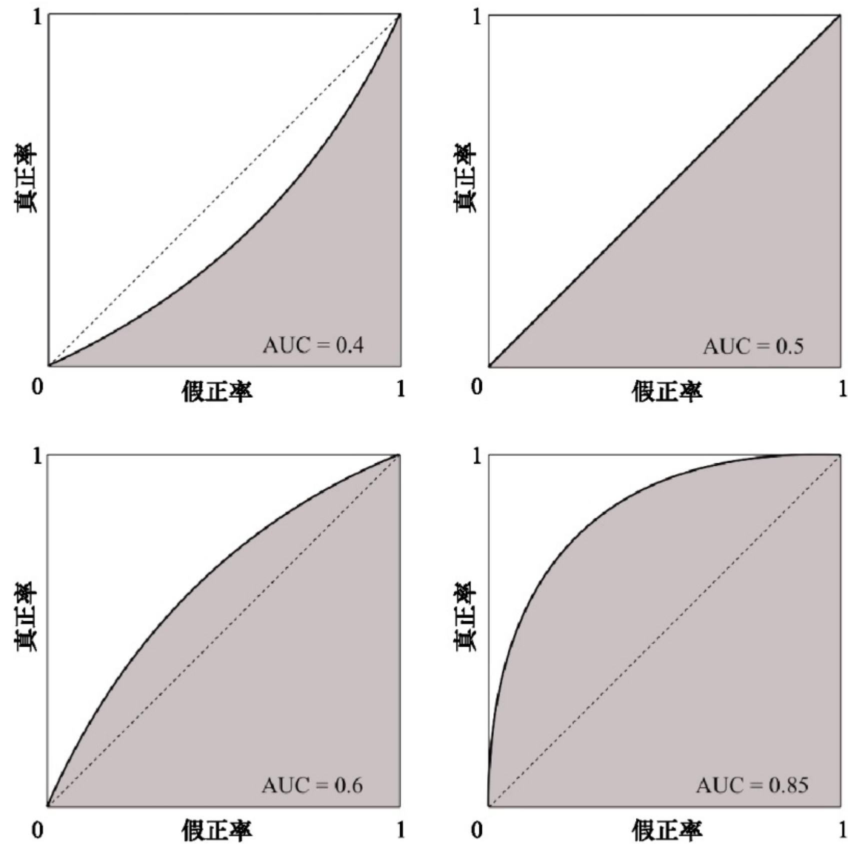
$$\text{真正率} = \frac{\text{正例}}{\text{正例} + \text{假负例}}$$

$$\text{假正率} = \frac{\text{假正例}}{\text{正例} + \text{假负例}}$$

ROC□□□□□□□□□□□□□□□(confidence)(□□□□□)□□□□□□□,□□□
□□□□□□□□□□(□□□□□□□□□□□□□)□□□ROC□□□□□

ROC□□□□□□:□□□□□□□□□□□□□□□□□□□□□□[0,1],□□□□□□□□□:
[0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]□□□,□□□□□□□□□□□□□□,
□□□□□□□□□□□□□□□□□□,□□□□□□□□□□0.7□□□□□□□□□,□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□0.7,□□□□□□□;□□,□□□□□□

□□5.3□□,□□□□0□,□□□□□□□□□,□□□□□□□□□□□1□□□□□,□□□□□1,
□□□□□□□□,□□□□□□□□□0,□□□□□□□□□□



□5.3 ROC□□□□□(□□)

ROC□□□□□(AUC)□□,□□□□□□□□□□AUC□□0.5□□□□□□□□□□□□□□□□AUC□□0.5,□□□□□□□□□□□□□□□□□□□□□□AUC□□□□1□□□□□□,□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□1,□□□□□□□□0□

ROC□□□□□□,□□□□□□□□□,□□□□□□□□□(□□□□□□□□□□□)□□□□□□□□□□□□□□□□□□□□□

5.7 □□□□□

□□□□□□□□□□□□□,□□□□□□□,□□□□□□□□□□□□□□□□□□ID3□□ϵ□d□SVM□□C□□□□□□□□α□□□□□□□□□?□□□□□□□□□?□□□□□□□□□□□□?□□□□□,□□□□□□□□□□

□□□□□□□□□,□□□□□□□□□□□□□□□□□□□“□□”□□□,□□□□□□□□□□□□□□□,□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□□□□□),□□□□□□□□□□□□□□□□,□□□□□□□□□□□□(grid search)□

□□□□□□□□□□□□□□□□□□□□□SVM□□,□□□□□□□□□□□:□□□□C(□□□□)□□□□[“linear”(“□□”)□“RBF”]□

□□□□□□□□□□□□□□,□□□□C□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□C□:[0.001,0.01,0.1,1,10,100,1000]□□□□□,□□□□□14□□□□□□□□□□□□□:[(0.001,“linear”),(0.01,“linear”),(0.1,“linear”),(1,“linear”),(10,“linear”),(100,“linear”),(1000,“linear”),(0.001,“rbf”),(0.01,“rbf”),(0.1,“rbf”),(1,“rbf”),(10,“rbf”),(100,“rbf”),(1000,“rbf”)]□

□□□□□,□□□□□□14□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□(□□□□□□□□□□)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□

[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible](random search)[illegible](Bayesian hyperparameter optimization)[illegible]

[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible](gradient-based technique)[illegible](evolutionary optimization technique)[illegible],[illegible],[illegible],[illegible]

[illegible]

[illegible],[illegible]
(cross-validation)[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible](fold)[illegible]5[illegible]5[illegible],[illegible]5[illegible]:{$F_1,F_2,\ldots,F_5$}[illegible]$F_k,k=1,\ldots,5$[illegible]20%[illegible],[illegible]5[illegible]f_1[illegible],[illegible]F_2,F_3,F_4,F_5[illegible],[illegible]F_1[illegible]f_2[illegible],[illegible]F_1,F_3,F_4,F_5[illegible],[illegible]F_2[illegible],[illegible]F_1[illegible]F_5[illegible]5[illegible]

[illegible],[illegible],[illegible],[illegible]

---

[1] [illegible],[illegible],[illegible],[illegible]{[illegible]},[illegible],[illegible]{4,3,2,1}[illegible]

[2] [illegible]k[illegible],[illegible]

[3] [illegible]

[illegible]6[illegible] [illegible]

[illegible],[illegible],[illegible],[illegible](logistic regression)![illegible],[illegible]([illegible])[illegible]softmax[illegible],[illegible]

6.1 [illegible]

[illegible],[illegible]

[illegible]SVM[illegible],[illegible](neural network,NN)[illegible]

$$y = f_{NN}(\boldsymbol{x})$$

[illegible]f_{NN}[illegible]:[illegible](nested function)[illegible](layer)[illegible]3[illegible]f_{NN},[illegible]:

$$y = f_{NN}(\boldsymbol{x}) = f_3(\boldsymbol{f}_2(\boldsymbol{f}_1(\boldsymbol{x})))$$

[illegible],$\boldsymbol{f}_1$[illegible]$\boldsymbol{f}_2$[illegible]:

$$f_l(\boldsymbol{z}) \stackrel{\text{def}}{=} \boldsymbol{g}_l(\boldsymbol{W}_l \boldsymbol{z} + \boldsymbol{b}_l) \tag{6.1}$$

□□,l□□□□,□□□1□□□□□□□□□□□□□$\boldsymbol{g}_l$□□□□□□(activation function)□□□□□□□□□□,□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□$\boldsymbol{W}_l$(□□□□)□$\boldsymbol{b}_l$(□□□□),□□□□□□□□□□□□□□□□□□(□□□□□)□□□□□□□□□□□□□□□□□□□□6.1□□□□□□□□,□□□$\boldsymbol{g}_l$□□□sigmoid□□,□□□□□□□□f_3□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□

□□□□□□□,□□□□□□□□□$\boldsymbol{W}_l$,□□□□□□□$\boldsymbol{w}_l$□□□□□$\boldsymbol{g}_l$□□□□□□□□$\boldsymbol{W}_l$□□□□$\boldsymbol{w}_{l,u}$(u□□□□)□□□□$\boldsymbol{z}$□□□□□□□□□$a_{l,u}=\boldsymbol{w}_{l,u}\boldsymbol{z}+b_{l,u}$,$\boldsymbol{f}_l(\boldsymbol{z})$□□□□□□$[g_l(a_{l,1}),g_l(a_{l,2}),\ldots,g_l(a_{l,\mathrm{size}_l})]$□□□,$g_l$□□□□□□□[1],$\mathrm{size}_l$□$l$□□□□□□□□□□□□□□,□□□□□□□□(multilayer perceptron,MLP)□□□□□□□□□□□□□□□□□(vanilla neural network)□

6.1.1 □□□□□□

□□□□□□□□□□□□□□□MLP□□□□□□□□□□□,□□□□□□□(feedforward neural network,FFNN)□□□□□□3□MLP□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□,□□□□□□□□,□□□□3□(□□□)□□□□□□□□

□6.1□□□□□MLP□□□□□□□□6.1□,□□□□□□(layer)□□□□□□□□,□□□□□□□□□□(□□□□,unit)□□

□6.1　□□□□□□□□□□□□

□□□□□□□□□4□□□□□□□□□□1□□□□

　　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□

　　□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□
□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□g,□□□□□□
□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□

　　□□6.1□,□□□□g_l□□□□□□l,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□
□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□$\boldsymbol{w}_{l,u}$□$b_{l,u}$,□□,l□u□
□□□□□□□□□□□□□□□,□□$\boldsymbol{y}_{l\text{-}1}$□□□□$[y_{l-1}^{(1)}, y_{l-1}^{(2)}, y_{l-1}^{(3)}, y_{l-1}^{(4)}]$□□□□□□□□$\boldsymbol{x}$□□□□
$[x^{(1)},\ldots,x^{(D)}]$□

　　□□6.1□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□(fully-connected layer)□□□□□□□□□□□□□□□□□□□□□□□□□□□

6.1.2　□□□□□

[illegible],[illegible]([illegible])[illegible]g_{last}[illegible],[illegible]g_{last}[illegible],[illegible]

[illegible][2][illegible]$g_{l,u}$[illegible]l[illegible]u[illegible]$\boldsymbol{w}_{l,u}$[illegible]$b_{l,u}$[illegible],[illegible]f_{NN}[illegible],[illegible],f_{NN}[illegible],$\boldsymbol{W}_l z+\boldsymbol{b}_l$[illegible],[illegible]

[illegible],[illegible]**TanH**[illegible]**ReLU**[illegible],[illegible],[illegible]-1[illegible]1([illegible]-1[illegible]1)[illegible],[illegible]z[illegible]0,[illegible]z[illegible]

$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}},$$

$$relu(z) = \begin{cases} 0 & z < 0 \\ z & \text{其他} \end{cases}$$

[illegible],$\boldsymbol{W}_l z+\boldsymbol{b}_l$[illegible]$\boldsymbol{W}_l$[illegible]$\boldsymbol{b}_l$[illegible]$\boldsymbol{w}z+b$[illegible]$\boldsymbol{W}_l$[illegible],[illegible]$u$[illegible]$\boldsymbol{w}_{l,u}$[illegible]$\boldsymbol{w}_{l,u}$[illegible]$l$-1[illegible]$\boldsymbol{W}_l z$[illegible]$\boldsymbol{a}_l \stackrel{\text{def}}{=} [\boldsymbol{w}_{l,1}z, \boldsymbol{w}_{l,2}z, \cdots, \boldsymbol{w}_{l,\text{size}_l}z]$[illegible],$\boldsymbol{a}_l+\boldsymbol{b}_l$[illegible]$\text{size}_l$[illegible]$\boldsymbol{c}_l$[illegible],[illegible]$g_l(\boldsymbol{c}_l)$[illegible]$\boldsymbol{y}_l \stackrel{\text{def}}{=} [y_l^{(1)}, y_l^{(2)}, \cdots, y_l^{(\text{size}_l)}]$[illegible]

6.2 [illegible]

[illegible],[illegible],[illegible],[illegible],[illegible](exploding

gradient)[illegible](vanishing gradient)[illegible]

[illegible],[illegible](gradient clipping)[illegible]L1[illegible]L2[illegible]

[illegible],[illegible]?[illegible]?[illegible],[illegible](backpropagation)[illegible],[illegible]4[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible](0,1)[illegible]n[illegible],[illegible]([illegible])[illegible]n[illegible],[illegible]n[illegible],[illegible]

[illegible],[illegible]([illegible])[illegible],[illegible]ReLU[illegible]LSTM([illegible],[illegible])[illegible](residual neural network)[illegible](skip connection),[illegible]

[illegible],[illegible]([illegible])[illegible]“[illegible]”[illegible],[illegible],[illegible](hidden layer)[illegible]

6.2.1 [illegible]

[illegible],MLP[illegible],[illegible]$(size_l+1)\cdot size_l$[illegible]([illegible]$\boldsymbol{W}_l$[illegible]$\boldsymbol{b}_l$)[illegible],[illegible]1 000[illegible],[illegible]100[illegible](computationally intensive)[illegible]

[illegible],[illegible][3][illegible]MLP[illegible],[illegible](intractable)[illegible]

[illegible](Convolutional Neural Network,CNN)[illegible]FFNN[illegible],[illegible]CNN[illegible],[illegible]CNN[illegible],[illegible]

[illegible]CNN[illegible],[illegible]

[illegible],[illegible]:[illegible]“[illegible]”[illegible]:[illegible]“[illegible]”[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible](patch)[4][illegible],[illegible],[illegible],[illegible](pattern)[illegible],[illegible],[illegible],[illegible]

在CNN中,卷积层的主要功能如图6.1所描述,以某种方式将1个矩阵变为2或3个矩阵。对于这个问题,我们需要知道一个被称为尺寸为$p\times p$的矩阵$\boldsymbol{F}$(过滤器,filter的中文翻译)的独立数学概念,p的值通常是一个小奇数。接下来以一个简单的图像为例,1和0分别表示白色和黑色像素。我们首先从该图像中提取出一个3×3大小(p=3)的一个子区域矩阵作为我们的小块$\boldsymbol{P}$(小块,patch的中文翻译):

$$
\boldsymbol{P} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}
$$

过滤器是一个我们所要寻找的、能够与某一特征相对应的模式(当然还有其他)。例如我们可以定义一个3×3的过滤器$\boldsymbol{F}$来检测图像中出现1的十字形的位置模式,其中0代表无、其他数字代表各种强度。为了寻找这些模式,我们将$\boldsymbol{P}$与$\boldsymbol{F}$做卷积(convolution),$\boldsymbol{F}$与$\boldsymbol{P}$越相似卷积运算得到的值就越高。假设我们有这样一个过滤器,我们定义$\boldsymbol{F}$为:

$$
\boldsymbol{F} = \begin{bmatrix} 0 & 2 & 3 \\ 2 & 4 & 1 \\ 0 & 3 & 0 \end{bmatrix}
$$

卷积运算的过程就是将两个矩阵中对应位置的元素,$\boldsymbol{P}$与$\boldsymbol{F}$相乘后再求和,如图6.2所示。

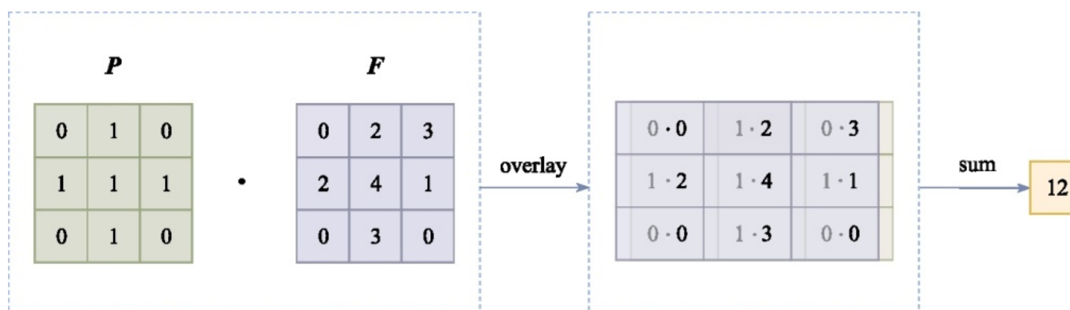


图6.2　卷积运算过程示意图

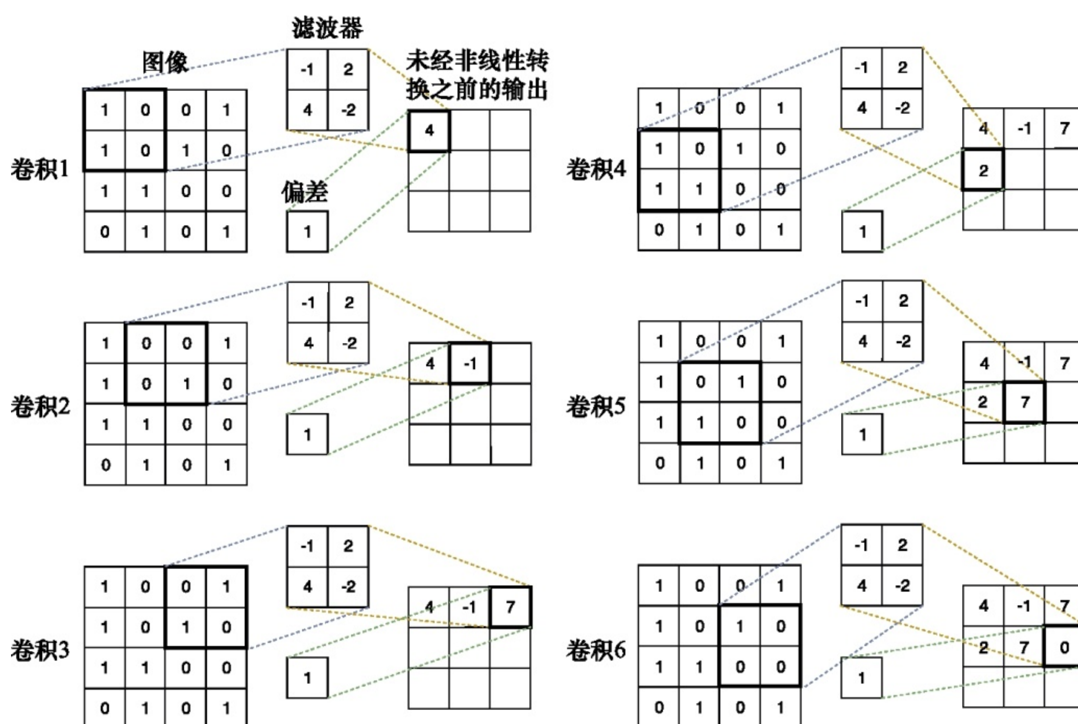
现在假设我们的$\boldsymbol{P}$是另外一个图像块,它表示的图案是字母L形:

$$
\boldsymbol{P} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}
$$

[illegible],[illegible]$\boldsymbol{F}$[illegible]:5[illegible],[illegible]“[illegible]”[illegible],[illegible][illegible],[illegible]$\boldsymbol{F}$[illegible]b,[illegible]([illegible])[illegible]

[illegible]CNN[illegible]([illegible]),[illegible]FFNN[illegible]([illegible])[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible]6.3[illegible]6[illegible]



[illegible]6.3 [illegible]

[illegible]([illegible])[illegible],[illegible]

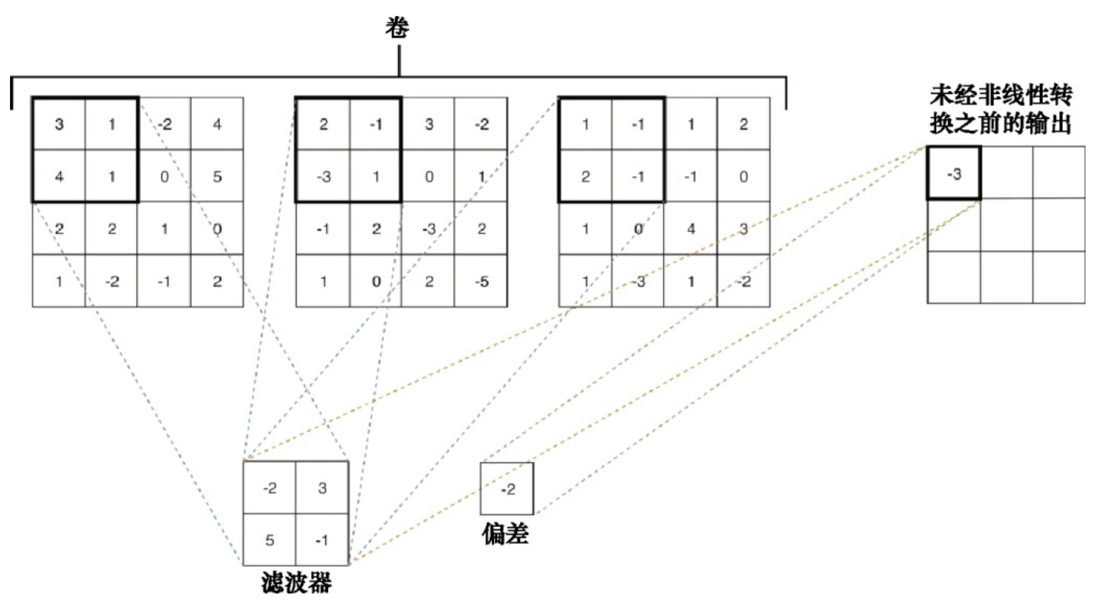
[illegible],[illegible],[illegible]ReLU[illegible]

[illegible]l[illegible]$size_l$[illegible],[illegible]$size_l$[illegible],[illegible]

□□CNN□□□□□□□□□□□□,□/+1□□□/□□□□□□size,□□□□□□□□□□□□□□□□□□□□□(volume),□□□□□□□□□□□□/+1□□□

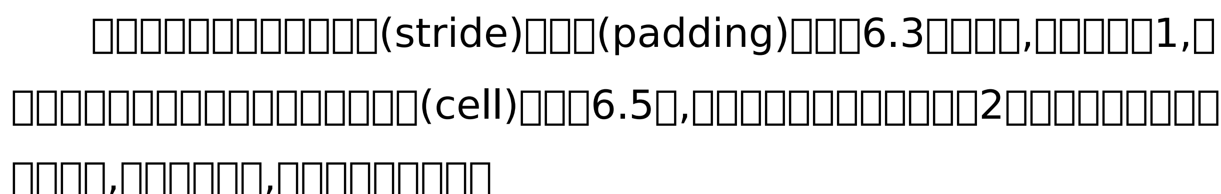
□□,□□□□□□□3□□□□□□□□□□□□□6.4□□□□□□□□□□□□:

[-2·3+3·1+5·4+(-1)·1]+[(-2)·2+3·(-1)+5·(-3)+(-1)·1]+[(-2)·1+3·(-1)+5·2+(-1)·(-1)]+(-2)=-3



□6.4 □□□□□□□□3□□□

□□□□□□□□□,□□□□3□□□(R□G□B)□□,□□□□□□□□□□□□□□□□□,CNN□□□□□□□□

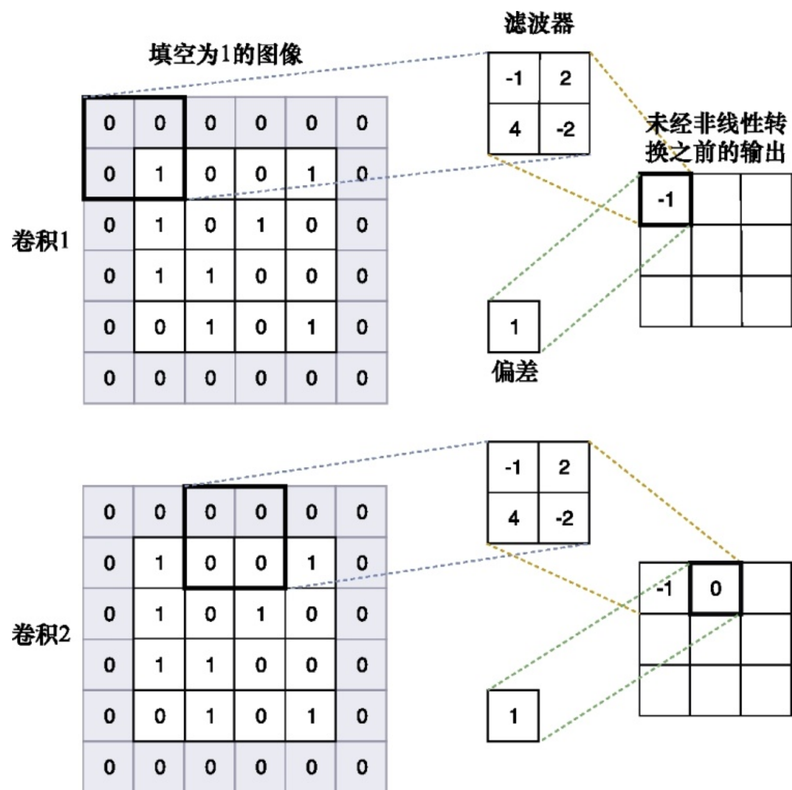


□□□□□□□□□□□□(stride)□□□(padding)□□□6.3□□□□,□□□□□1,□
□□□□□□□□□□□□□□□□□(cell)□□□6.5□,□□□□□□□□□□□□2□□□□□□□□□
□□□□,□□□□□□,□□□□□□□□□



□6.5 □□□2□□□

□□□,□□□□□□□;□□□□□□□□□□□□□□□□,□□□□□(□□)□□□□□□□□□□□
□□□□□□□□□□0□□□6.3□,□□□0,□□□□□□□□□□□□□□□□□□6.6□,□□□2,□
□□1,□□□□□□□□□□□□1□□□□□□□□,□□□□□□□,□□□□□□□[5]□



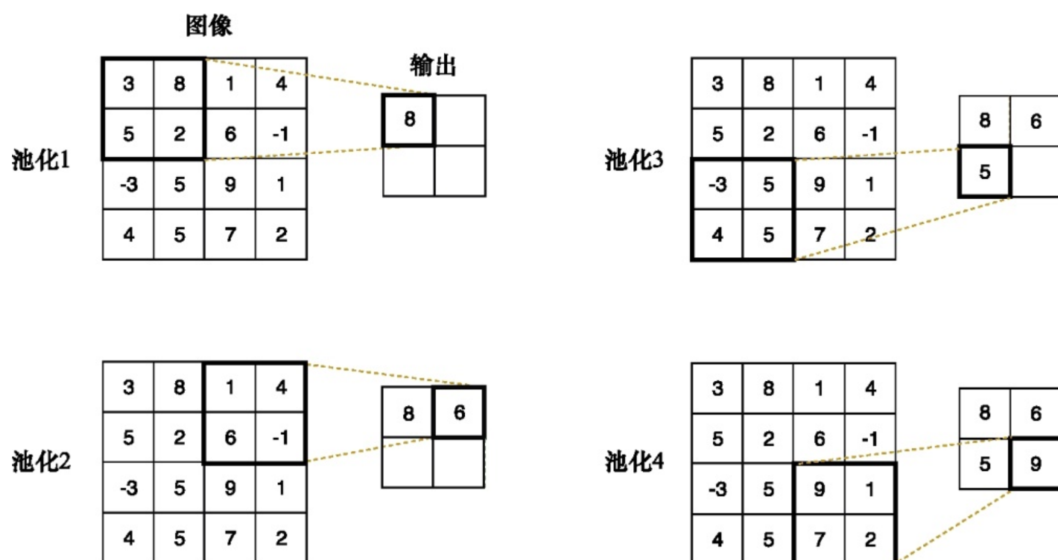
□6.6 □□□2□□□□1□□□

□6.7□□□□□□2□□□□□□□□□□□□□□□□□□□“□□”□□□□□

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0
0	0	1	0	1	0	0	0
0	0	1	1	0	0	0	0
0	0	0	1	0	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

□6.7 □□□2□□□

□□,□□□□□□□□□□(pooling),□□□□□□CNN□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□(max)□□□□(average)□□□□□□□□□□,□□□□□□□:□□□□□□□□□□□□□6.8□□,□□□□□□□□□□□□□□□□□2□



□6.8 □□□□□□2□□□□2□□□□□

□□□□□,□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□

□□□□,□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□2□3,□□□□2□□□□□□□□,□□□□□□□,□□□□□□□

□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□(□□6.8□□,□□□□□□□□□□□2□,□□□□□□□)□

6.2.2 □□□□□□

[illegible](Recurrent Neural Network,RNN)[illegible],[illegible],[illegible](
[illegible])[illegible]

RNN[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible]l[illegible]u[illegible](state)[illegible]$h_{l,u}$[illegible]RNN[illegible],[illegible]l[illegible]u[illegible]:[illegible]l-1[illegible],[illegible]l[illegible]

[illegible],[illegible]RNN[illegible]([illegible])[illegible]

[illegible]6.9[illegible],[illegible],[illegible],[illegible]$\boldsymbol{X}=[\boldsymbol{x}^1,\boldsymbol{x}^2,\ldots,\boldsymbol{x}^{t-1},\boldsymbol{x}^t,\boldsymbol{x}^{t+1},\ldots,x^{\mathrm{length}_x}$
],$\mathrm{length}_{\boldsymbol{x}}$[illegible]$\boldsymbol{X}$[illegible],[illegible]$\boldsymbol{x}^t$[illegible]$t$[illegible],$t=1,\ldots,\mathrm{length}_{\boldsymbol{x}}$[illegible]

□6.9　□□RNN□□□□

□□□□□□□□□□□□□□□□□□□□□□□

□□6.9□□,□RNN□,□□□□□□□□□□□□□□□□□□□□□(timestep)□□“□□”□□□t□□□□□□□□□□□□□□□□□□□□□□t,□□□l□□□□□u□□□□$h_{l,u}^{t}$□□□,□□□□□□□□□□□□□□□□□$\boldsymbol{h}_{l,u}^{t-1}$□□□□□□$\boldsymbol{h}_{l,u}^{t-1}$□□□□□□□□□□□$t$-1□□□□□□□□□□□□□□□□□□□□□□□□$\boldsymbol{w}_{l,u}$,$\boldsymbol{u}_{l,u}$□□□□□□$b_{l,u}$□□□□□□□,□□□□□□□□□□□□□□□□g_1,□□$\boldsymbol{h}_{l,u}^{t}$□□□□□□□,□□□□$tanh$□□□□□□g_1□□□$\boldsymbol{y}_{l}^{t}$□□□□□□□,□□□□□□□l□□□□□□$\boldsymbol{y}_{l}^{t}$,□□□□□□□□□□□□g_2□□□□□□□,g_2□□□□□□□□□□□□□□□□□□□□,□□□□□□□□$\boldsymbol{h}_{l,u}^{t}$□□□□□□□g_2□$\boldsymbol{h}_{l,u}^{t}$□□□□□□□$\boldsymbol{V}_l$□□□□□□□$\boldsymbol{c}_{l,u}$□□□□□□□□□□□□,□□□□**softmax**□□□□g_2:

$$\boldsymbol{\sigma}(z) \stackrel{\text{def}}{=} [\sigma^{(1)}, \cdots, \sigma^{(D)}], □□ \quad \boldsymbol{\sigma}^{(j)} \stackrel{\text{def}}{=} \frac{\exp\left(z^{(j)}\right)}{\sum_{k=1}^{D} \exp\left(z^{(k)}\right)}$$

softmax□□□sigmoid□□□□□□□□□□□□□□□□□□□□□:$\sum_{j=1}^{D}\sigma^{(j)}$=1,□□□□$j$□$\sigma^{(j)}$>0□

□□$\boldsymbol{V}_l$□□□□□□□□□□□□□$\boldsymbol{V}_l$□□□$\boldsymbol{h}_l^t$□□□□□□□□□□$\boldsymbol{c}_l$□□□□□□□□□□□□□□□□□□□□□□□□□□y□□□□(□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□)

□□□□□□□□□□$\boldsymbol{w}_{l,u}$,$\boldsymbol{u}_{l,u}$,$b_{l,u}$,$\boldsymbol{V}_{l,u}$□$\boldsymbol{c}_{l,u}$□□,□□□□□□□□□□□□□□□□RNN□□□□□□□□□□□□□——□□□□□□□□□(backpropagation through time)□

tanh□softmax□□□□□□□□□□□□□□,□□□□□□□□□□□□□RNN□□□□□□□□□□,□□□□□□□□□□□□“□□”□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□

RNN□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□RNN(gated RNN)□□□□□□□□□□□□(long short-term memory,LSTM)□□,□□□□□□□□□□□(gated recurrent unit,GRU)□□□□

□RNN□□□□□□(gated unit)□□□□,□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□(bit)□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□(0,1)□□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□□□□□,□□□□□□□□□t□□□□□□□□□□

[illegible],[illegible],[illegible],[illegible]RNN[illegible],[illegible]

[illegible],[illegible],[illegible](gate)[illegible]GRU(minimal gated GRU),[illegible]

[illegible]RNN[illegible]([illegible])[illegible]GRU[illegible],[illegible] l [illegible]GRU[illegible] u [illegible]:[illegible] $\boldsymbol{h}_l^{t-1}$ [illegible] $\boldsymbol{x}^t$ [illegible],[illegible]([illegible])[illegible]

$$
\begin{aligned}
\widetilde{h}^t_{l,u} &\leftarrow g_1(\boldsymbol{w}_{l,u}\boldsymbol{x}^t + \boldsymbol{u}_{l,u}\boldsymbol{h}_l^{t-1} + b_{l,u}),\\
\Gamma^t_{l,u} &\leftarrow g_2(\boldsymbol{m}_{l,u}\boldsymbol{x}^t + \boldsymbol{o}_{l,u}\boldsymbol{h}^{t-1} + a_{l,u}),\\
h^t_{l,u} &\leftarrow \Gamma^t_{l,u}\,\widetilde{h}^t_l + (1-\Gamma^t_{l,u})h_l^{t-1},\\
\boldsymbol{h}_l^t &\leftarrow [h^t_{l,1},\cdots,h^t_{l,\text{size}_l}]\\
\boldsymbol{y}_l^t &\leftarrow g_3(\boldsymbol{V}_l\boldsymbol{h}_l^t + \boldsymbol{c}_{l,u})
\end{aligned}
$$



[illegible], g_1 [illegible]tanh[illegible], g_2 [illegible](gate function),[illegible]sigmoid[illegible](0,1)[illegible] $\Gamma_{l,u}$ [illegible]0[illegible],[illegible] h_l^{t-1} [illegible],[illegible] $\Gamma_{l,u}$ [illegible]1[illegible],[illegible] $\widetilde{h}^t_{l,u}$ [illegible]([illegible]3[illegible])[illegible]RNN[illegible], g_3 [illegible]softmax[illegible]

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(identity function, $f(x)=x$)□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□

□RNN□□□□□□□□□□□□□□□□□□□□(bi-directional RNN),□□□□□(attention)□□□□□□□□□□□□□□□□□□□□(sequence-to-sequence RNN)□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□RNN□□□□□□□□□□(RecursiveNeuralNetwork)□

---

[1] □□□□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□

[2] □□□□□□□□□□□□□□□□□□□□□□□□□□,ReLU□□0□□□□□

[3] □□□□□□□□□□□□□□□□□□□□□100×100□□□,□□□10 000□□□□

[4] □□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□

[5] □□□□□,□6.6□□□□9□□□□□□□□□□□

□7□ □□□□□□□

7.1 □□□

□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□?□□□□□□□□□□□□□□□□□□□□□□□□$\{(x_i, y_i)\}_{i=1}^{N}$,□□□□□□□□□□□□□□□□□□□□□□□□□□(mean squared error,MSE),□□□□□□□□□,□□□□□□□□□□□□□w_1,w_2□□b□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□D□(D>3)□,□□□□□□□□□□□□□□□□□

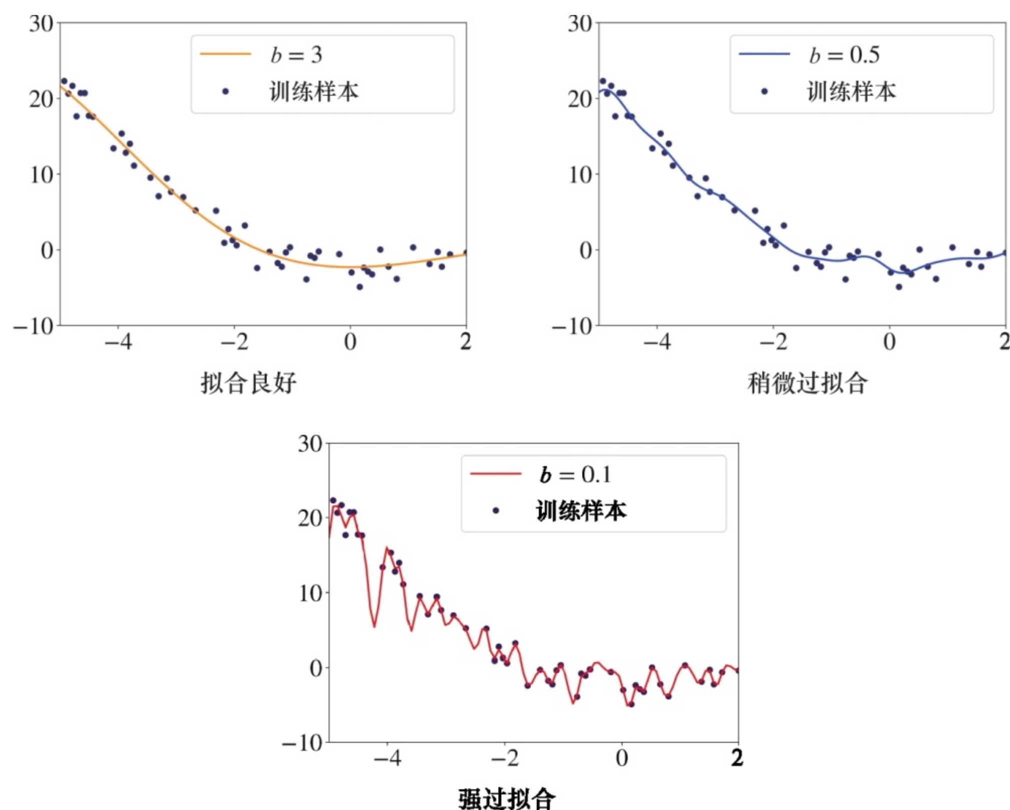
□□□□□□□□□□□(non-parametric method)□□□□□,□□□□□□□□□□□,□□□□□□□(□□□□kNN□□□□)□□□□□,□□□□□,□□□□□□□□□□:

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} w_i y_i, \quad □□ \quad w_i = \frac{N k\left(\frac{x_i - x}{b}\right)}{\sum_{l=1}^{N} k\left(\frac{x_l - x}{b}\right)} \tag{7.1}$$

□□$k(\bullet)$□□□(kernel)□□□□□□□□□□□□□:x□x_i□□□,□□w_i□□□;□□□□□□□□□□□□□□□□□,□□□□□□□□□(Gaussian)□:

$$k(z) = \frac{1}{\sqrt{2\pi}}\exp\left(\frac{-z^2}{2}\right)$$

[illegible]7.1[illegible],b[illegible],[illegible]([illegible]b[illegible],[illegible]MSE)[illegible]7.1,[illegible]b[illegible]



[illegible]7.1 [illegible]3[illegible]b[illegible]

[illegible],[illegible]7.1[illegible]x_i-x[illegible]$x_{[illegible]}$-x[illegible]$||\boldsymbol{x}_i-\boldsymbol{x}||$[illegible]$||\boldsymbol{x}_{[illegible]}-\boldsymbol{x}||$[illegible]

7.2 [illegible]

[illegible],[illegible],[illegible]

[illegible],[illegible]C[illegible]:$y \in \{1,\dots,C\}$[illegible],[illegible]SVM[illegible]ID3[illegible]:

$$f_{\mathrm{ID3}}^{S} \stackrel{\mathrm{def}}{=} \Pr(y_i = c|\boldsymbol{x}) = \frac{1}{|S|}\sum_{\{y|(\boldsymbol{x},y)\in S, y=c\}} y$$

[illegible],c[illegible]$c \in \{1,\dots,C\}$,S[illegible]

[illegible],[illegible]sigmoid[illegible]**softmax**[illegible],[illegible]6[illegible],[illegible]softmax[illegible]

kNN[illegible]:[illegible]$\boldsymbol{x}$[illegible]k[illegible],[illegible],[illegible]

SVM[illegible],[illegible]?[illegible](one versus rest)[illegible],[illegible]C[illegible],[illegible]C[illegible],[illegible]3[illegible]$y \in \{1,2,3\}$,[illegible],[illegible]1[illegible]0[illegible],[illegible]2[illegible]0[illegible],[illegible]3[illegible]0[illegible],[illegible]3[illegible],[illegible]1[illegible]0[illegible]2[illegible]0[illegible]3[illegible]0[illegible]

[illegible]3[illegible],[illegible]$\boldsymbol{x}$[illegible],[illegible],[illegible]3[illegible],[illegible](the most certain)[illegible]0[illegible],[illegible](0[illegible]1[illegible])[illegible],[illegible](certainty)[illegible]SVM[illegible],[illegible]$\boldsymbol{x}$[illegible]d,[illegible]:

$$d \stackrel{\text{def}}{=} \frac{\boldsymbol{w}^* \boldsymbol{x} + b^*}{||\boldsymbol{w}||}$$

[illegible],[illegible],[illegible]

7.3 [illegible]

[illegible],[illegible],[illegible]

[illegible](one-class classification)[illegible](unary classification)[illegible](class modeling)[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible](outlier detection,anomaly detection)[illegible](novelty detection)[illegible]

[illegible],[illegible](one-class Gaussian)[illegible]**k**[illegible]**kNN**[illegible]**SVM**[illegible]

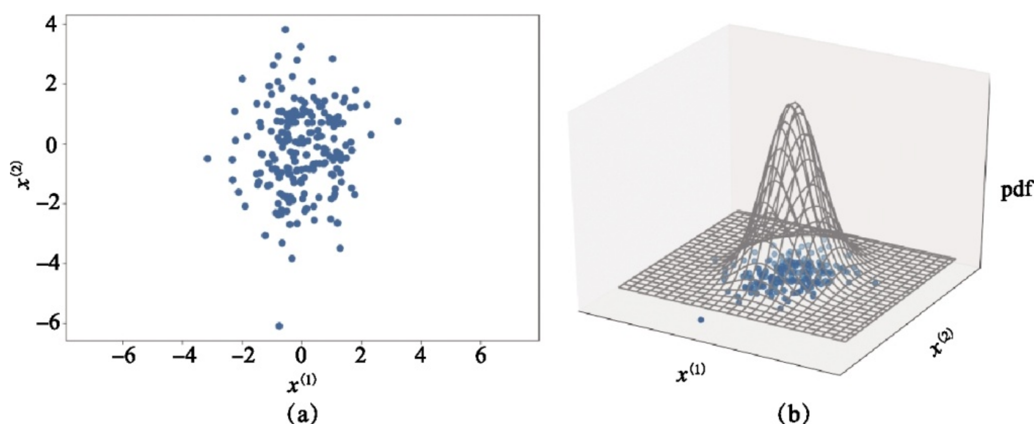
[illegible],[illegible],[illegible](multivariate normal distribution,MND),[illegible]MND[illegible](probability density function,pdf)[illegible]:

$$f_{\boldsymbol{\mu},\boldsymbol{\Sigma}}(\boldsymbol{x}) = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}}$$

□□,$f_{\mu,\boldsymbol{\Sigma}}(\boldsymbol{x})$□□□□□□□$\boldsymbol{x}$□□□□□□□□□□□□□□□□□□□□$\boldsymbol{x}$□□□□□□□□□□□□□,□□□MND□□□□□□□$\boldsymbol{\mu}$(□□□□)□$\boldsymbol{\Sigma}$(□□□□)□□□□□□□□□□□□□□□□□□□□(maximum likelihood)□□(□□□□□□□□□□□□□)□□□□□□□□□□□□ $|\boldsymbol{\Sigma}| \stackrel{\text{def}}{=} \det \boldsymbol{\Sigma}$,□□□$\boldsymbol{\Sigma}$□□□□(determinant);$\boldsymbol{\Sigma}^{-1}$□□$\boldsymbol{\Sigma}$□□□□(inverse)□

□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□(matrix theory)□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□

□□□□,□□$\boldsymbol{\mu}$□□□□□□□□□□□□□□□□□□□□□□$\boldsymbol{\Sigma}$□□□□□□□□□□□□□□7.2□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□



□7.2 □□□□□□□□□□□□□□□□□□

□:(a)□□□□□□;(b)□□7.2(a)□□□□□□□□□MND□□□

□□□□□□□□□□□□□$\boldsymbol{\mu}$□$\boldsymbol{\Sigma}$□,□□□□$f_{\mu,\boldsymbol{\Sigma}}(\boldsymbol{x})$□□□□□□$\boldsymbol{x}$□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□;□□,□□□□□□□□□□□□□□□□□□□,□□□“□□□□□□(educated guess)”□□□

[illegible],[illegible]([illegible])[illegible],[illegible]:[illegible]1[illegible]$\boldsymbol{\mu}$[illegible]1[illegible]$\boldsymbol{\Sigma}$,[illegible]9[illegible],[illegible]

[illegible]k[illegible]kNN[illegible]:[illegible],[illegible]k[illegible],[illegible]**k**[illegible](k-mean)[illegible]$\boldsymbol{x}$,[illegible]$d(\boldsymbol{x})$[illegible]$\boldsymbol{x}$[illegible],[illegible]$d(\boldsymbol{x})$[illegible],[illegible]$\boldsymbol{x}$[illegible]

[illegible]SVM,[illegible],[illegible]([illegible])[illegible],[illegible],[illegible],[illegible](hypersphere)[illegible]kNN[illegible],[illegible]k[illegible]SVM[illegible],[illegible]

7.4 [illegible]

[illegible],[illegible](multi-label classification)[illegible]

[illegible],[illegible]7.3[illegible],[illegible]:“[illegible]”“[illegible]”“[illegible]”[illegible]



($y_{i,l}$∈{0,1}),□□l=1,...,L□i=1,...,N□□□□□$\boldsymbol{x}$□□□□l□□□$\hat{y}_{i,l}$,□□□□□□□□
□:

$$-\left(y_{i,l}\ln\left(\hat{y}_{i,l}\right)+\left(1-y_{i,l}\right)\ln\left(1-\hat{y}_{i,l}\right)\right)$$

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□,□□□,□□□□□□□□□□,□□□□□□:□□□□□□□□□□□,□{□□□□□};□□□□□□□□□□{□□□□□□□□}□□□□□□□□□□□□□□□□□□□□□□,□□□□:

□□□	□□□□1	□□□□2
1	□□	□□
2	□□	□□
3	□□	□□
4	□□	□□
5	□□	□□
6	□□	□□

[illegible],[illegible],[illegible]1[illegible]6[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

7.5 [illegible]

[illegible]3[illegible],[illegible],[illegible](ensemble learning)[illegible]

[illegible],[illegible],[illegible],[illegible](meta-model)[illegible]

[illegible](weak learner)[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible]***x***[illegible],[illegible],[illegible]:[illegible],[illegible]***x***[illegible]

[illegible](boosting)[illegible](bagging)[illegible]

7.5.1 集成学习的基本思想

集成学习是一种通过训练大量低精度模型并组合其预测结果来构建高精度模型的学习范式,其中的低精度模型被称为“弱学习器”。训练弱学习器的速度很快,而且多个弱学习器组合起来通常可以得到比单个强模型更好的集成模型(ensemble model)。下面介绍两种常见的集成学习方法。

第一种是套袋法,它使用原始数据的“副本”(从原始数据中重复抽样得到)来训练,每个副本训练一个弱模型并将其结果组合起来。第二种是提升法,最常见的套袋法是随机森林(random forest)。提升法依次训练一系列弱模型,每个新的模型都试图修正前一个模型的错误。

7.5.2 随机森林

传统的套袋法的原理如下。给定一个训练集,我们创建B个随机样本S_b(b=1,…,B),并用S_b训练一个决策树模型得到模型f_b。为了得到样本S_b集,我们采用有放回抽样(sampling with replacement)的方法,即保持一个空集合,然后随机从训练集中选取样本放入集合S_b,已选的样本可能被再次选中直到满足条件,直到$|S_b|=N$为止。

训练结束后,我们得到了B个决策树。对于新样本$\boldsymbol{x}$的预测结果为B个预测的平均值:

$$y \leftarrow \hat{f}(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{B}\sum_{b=1}^{B} f_b(\boldsymbol{x})$$

对于分类任务,则采用多数投票。

随机森林与传统的套袋法只有一个不同。它使用一种改进的树学习算法,在学习过程中的每次分裂时都检查特征的一个随机子集。这样做的原因是为了避免树之间的相关性:如果某个或某些特征对预测结果很重要,那么它们会在很多树中被选中用于分裂数据。

[illegible]“[illegible]”[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible] B [illegible]

[illegible]?[illegible],[illegible],[illegible](variance)[illegible](overfitting)[illegible],[illegible],[illegible],[illegible],illegible[illegible]:[illegible],[illegible],[illegible]

7.5.3 [illegible]

[illegible](gradient boosting)[illegible],[illegible] $f=f_0$ [illegible]([illegible]ID3[illegible]):

$$f = f_0(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{N}\sum_{i=1}^{N} y_i$$

[illegible],[illegible] $i=1,\ldots,N$ [illegible]:

$$\hat{y}_i \leftarrow y_i - f(\boldsymbol{x}_i) \tag{7.2}$$

[illegible],$\hat{y}_i$ [illegible](residual),[illegible] $\boldsymbol{x}$ [illegible]

假设我们采用的是当前训练的模型,它被新模型进行了加强。我们用f_1来表示,则新的增强模型为$f \stackrel{\text{def}}{=} f_0 + \alpha f_1$,其中$\alpha$为学习速率(超参数)。

接下来,我们用式7.2计算新的残差,并且再次将训练集中的标签替换为这些新的残差,创建新的决策树模型f_2,并将增强模型定义为$f \stackrel{\text{def}}{=} f_0 + \alpha f_1 + \alpha f_2$。重复该过程,直到组合树达到预定义的最大数量$M$。

直觉上这是为什么?通过计算残差,我们找到了目标与当前模型f预测之间的偏差值(或残差值)的大小。然后我们训练,使下一个决策树能修复当前模型的错误(这也是我们使用残差的原因),并将其与α值加权后添加到现有模型中。因此,每个添加到模型中的新的决策树都会部分修复之前决策树的错误,直到组合达到预定义的最大数量M(决策树数量)。

那么现在你可能会问一个问题?在第4章我们讨论过梯度下降,而这里的梯度在哪里,为何这个算法被称为梯度提升呢?在梯度提升中没有计算梯度,而是使用残差作为梯度的代理:它告诉我们如何调整模型来减小误差(残差)。梯度提升的三个主要超参数是决策树的数量、学习速率和树的深度,这三者都会影响模型的准确率。另外树的深度也会影响训练和预测的速度:深度越小,速度越快。

可以看出,通过残差训练可以优化整个模型f的均方误差。可以看出它与装袋法的区别:提升降低了偏差(或欠拟合)而非方差。正因如此,提升可能会导致过拟合。但是通过调整树的深度和数量,可以在很大程度上避免过拟合。

梯度提升在分类与回归问题中非常相似,但步骤稍有不同。我们考虑二分类问题,假设有M个回归决策树。与逻辑回归类似,我们使用sigmoid函数,将决策树的预测为:

$$\Pr(y=1|\boldsymbol{x},f)\overset{\text{def}}{=}\frac{1}{1+\mathrm{e}^{-f(\boldsymbol{x})}}$$

□□, $f(\boldsymbol{x})\overset{\text{def}}{=}\sum_{m=1}^{M}f_m(\boldsymbol{x})$, f_m□□□□□□□

□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□ $L_f=\sum_{i=1}^{N}\ln[\Pr(y_i=1|\boldsymbol{x}_i,f)]$ □f□□□,□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□

□□,□□□□□□□□□□ $f=f_0=\frac{p}{1-p}$,□□ $p=\frac{1}{N}\sum_{i=1}^{N}y_i$ (□□□□□□□□□□sigmoid□□□□□□)□□□,□□□□□m□,□□□□f_m□□□□□□□□□□□□f_m,□□□□i=1,…,N□□□□□□□□□g_i:

$$g_i=\frac{\mathrm{d}L_f}{\mathrm{d}f}$$

□□,f□□□□□□□m-1□□□□□□□□□□□□□g_i,□□□□□ln[Pr(y_i=1|$\boldsymbol{x}_i$,f)]□f□□□□i□□□□□□□□□□,□ln[Pr(y_i=1|$\boldsymbol{x}_i$,f)] $\overset{\text{def}}{=}\ln\left[\frac{1}{1+\mathrm{e}^{-f(\boldsymbol{x}_i)}}\right]$□,□□□□□□$f$□□□□□ $\frac{1}{\mathrm{e}^{f(\boldsymbol{x}_i)}+1}$□

□□,□□□□□□□,□□□y_i□□□□□□□□□g_i,□□□□□□□□□□□□f_m□□□□□□□□□□□□□□ρ_m:

$$\rho_m\leftarrow\arg\max_{\rho}L_{f+\rho f_m}$$

□□□m□□□,□□□□□□□□f,□□□□f_m:

$$f \leftarrow f + \alpha \rho_m f_m$$

□m=M□,□□□□,□□□□□□□f□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□

7.6 □□□□□□

□□□□□□□□□□□□□□□:□□□□□□□□□□□□□;□□□□□□□□;□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□

□□□□(sequence labeling)□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□($\boldsymbol{X}$,$\boldsymbol{Y}$)□□□,$\boldsymbol{X}$□□□□□□,□□□□□□□□□(timestep);$\boldsymbol{Y}$□□□□□□□□□□,$\boldsymbol{X}$□□□□□□□□□□,□[“□”“□□”“□□”],$\boldsymbol{Y}$□□□□□□□(part of speech),□[“□□□”“□□□”“□□”]□□□□□□□□,□□□□□i□,$\boldsymbol{X}_i$=[$\boldsymbol{x}_i^1$,$\boldsymbol{x}_i^2$,…,$\boldsymbol{x}_i^{\text{size}_i}$]□□□,size$_i$□□□$i$□□□□□,$\boldsymbol{Y}_i$=[$y_i^1$,$y_i^2$,…,$y_i^{\text{size}_i}$]□$y_i$∈{1,2,…,$C$}□

□□□RNN□□□□□□,□□□□□□□□□□□□□□□□□□t,RNN□□□□□□□□□$\boldsymbol{x}_i^{(t)}$,□□□□□□□□□□□□□□$\boldsymbol{y}_{\text{last}}^{(t)}$(□□□□□)□$y_{\text{last}}^{(t)}$(□□□□□□□□)□

□□,RNN□□□□□□□□□□□□□□□,□□□□(Conditional Random Field,CRF)□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□(named entity extraction)□□,□□□□□□□□□,□□□□□□[□“I go to San Francisco”(□□□□□)]□□□□□□□□□□□□:{□□

(location),□□(name),□□□□(company name),□□(other)}□□□□□□
□□□□(□□□)□□□□□□,□“□□□□□□□□□□□”□“□□□□□□□□□□□”,□□□□□□□□
□□□□□□□□□□□□□□□



□□□□□□□□□□,□□□□□□□□□□□□□□□

CRF□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□,□□□□
□□□□□□□□□RNN□□□□□□□,□□□□□□□□□□□,□□CRF□□□□□□□□□(□□□□□□
□□)□□□□□,□□□□□□□□□□□□□□□□□

7.7 □□□□□□□

□□□□□□□(sequence-to-sequence learning,seq2seq)□□□□□
□□□□□□□□□seq2seq□,X_i□Y_i□□□□□□□□seq2seq□□□□□□□□□□(□□□
□□□□□,□□□□□□□□□□□)□□□□□(□□□□□□□,□□□□□)□□□□□□□□□□□□

□□□□□□,□□□□□□□□□□seq2seq□□□□□□□□□□seq2seq□□□□□□□
□□□:□□□□□(encoder)□□□□□□(decoder)□

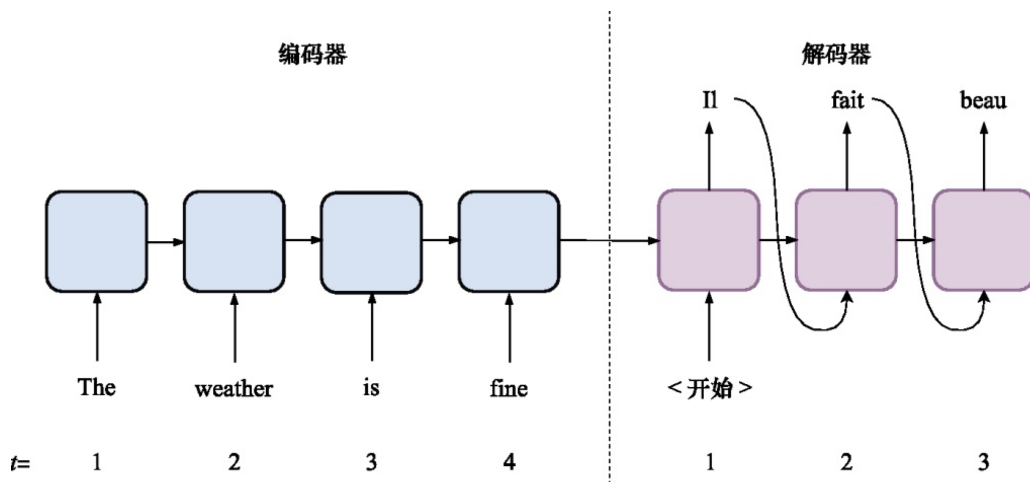
□seq2seq□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□□RNN,□□□□
CNN□□□□□□□□□□□□□□□□□□□□□□□□(□RNN□□□□□□)□□□□□□□□□□□
□□□□□□,□□□□□□□□□□□□(□□□□□□□□□□□□□□□□)□□□□□□□□□□□□□□□
□□□□□□□□□□□□□,□□□□(□□□)□□□□□□□(embedding)□

□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□
□□□□□□□□□□□□□□,□□□□□□□□□□□$\boldsymbol{x}^{(0)}$□□□□□(start of sequence)
(□□□□□0),□□□□□□□$\boldsymbol{y}^{(1)}$□□□,□□□□□□$\boldsymbol{x}^{(0)}$□□□□□□□□□□□□□,□□□$\boldsymbol{y}^{(1)}$□
□□□□□$\boldsymbol{x}^{(1)}$□□□□□□,□□□$\boldsymbol{y}^{(t)}$□$\boldsymbol{x}^{(t)}$□□□□□;□□,□□□□□□□□□□□□□□6□□
□,RNN□□□□□□□□□□□□□□□:□□□□□□□□□□$\boldsymbol{y}^{(t)}$,□□□□□□□□□□□□□$\boldsymbol{x}^{(t)}$□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□



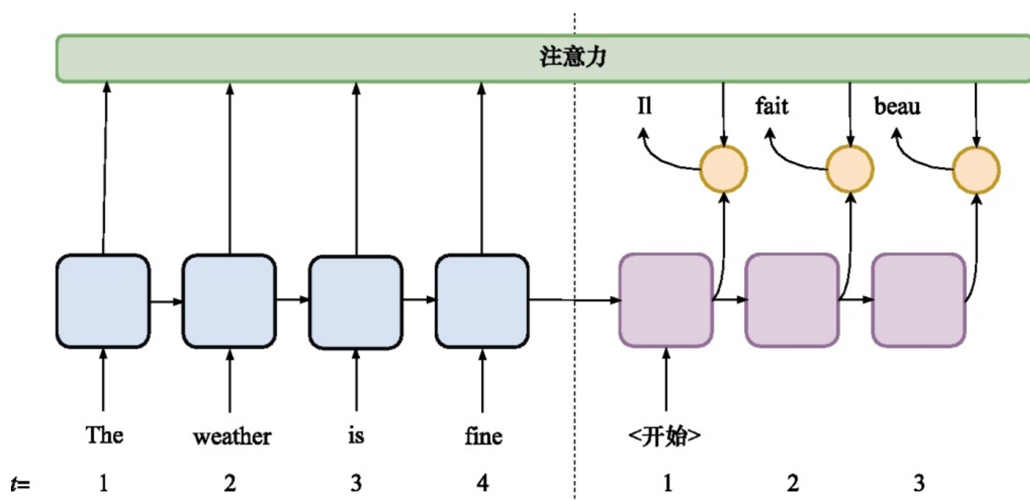
□□□□□seq2seq□□□□7.4□□□□□□□□□□□□□□□□□□(attention)□□
□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□(□RNN□,□□□□□□□□□□□□□
□□□□□□□□□□□)□□□□□□□□□□□□□□□□□□□□□□RNN□□,□□□□□□□□□□□
□□□



□7.4　□□□□□seq2seq□□

□:□□,□□□□□□□□□□□□□□□□□□□□7.4□,□□□□□□□□□□□□□□□

□7.5□□□□□□□□□□□seq2seq□□□



□7.5　□□□□□□□□seq2seq□□

seq2seq□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□

□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□

7.8 □□□□

□□□□(active learning)□□□□□□□□□□□□,□□□□□□□□□□□□□□□□
□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□
□,□□□□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□
□□

□□□□□□□□□□,□□□□□□□□□□□□:

- □□□□□□□□□□□□
- □□□□□□□

□□□□□□,□□□□□□□□□□□□□□□□f,□□□□□□□□□□□□□□(□□□□□□□□,
□□□□□□□□□)□□□□□□□□□□□$\boldsymbol{x}$□□□□□□□:□□($\boldsymbol{x}$)•□□□□$f(\boldsymbol{x})$□□□□□□$\boldsymbol{x}$□□
□□□□□□□□,□□□□□□□□□f□$\boldsymbol{x}$□□□□□□□□□□□□□□sigmoid□□□□□□□,□□
□□□□□0.5,□□□□□□□□□SVM□,□□□□□□□□□,□□□□□□□□

□□□□□□□,□□□□□□□□□□□□□□□:

$$H_f(\boldsymbol{x}) = -\sum_{c=1}^{C} \Pr\left(y^{(c)};\quad f(\boldsymbol{x})\right) \ln\left[\Pr\left(y^{(c)};\quad f(\boldsymbol{x})\right)\right]$$

□□,$\Pr(y^{(c)};f(\boldsymbol{x}))$□□□$f$□□□$y^{(c)}$□□□$\boldsymbol{x}$□□□□□□□□□□,□□□□□□
$y^{(c)}$,$f(y^{(c)})=\frac{1}{C}$,□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□$y^{(c)}$,$f(y^{(c)})=1$,
□□□□□□□□□□$y^{(c)}$,□□0□

□□$\boldsymbol{x}$□□□□□□□□□$\boldsymbol{x}$□k□□□□□□□□□□□□□□(k□□□□□□)□

[illegible],[illegible],[illegible],[illegible]([illegible],[illegible])[illegible]

[illegible],[illegible]SVM[illegible],[illegible],[illegible],[illegible],[illegible]([illegible])[illegible]



[illegible],[illegible]“[illegible]”(query by committee)[illegible],[illegible]

7.9 [illegible]

[illegible](semi-supervised learning,SSL)[illegible],[illegible];[illegible],[illegible],[illegible]

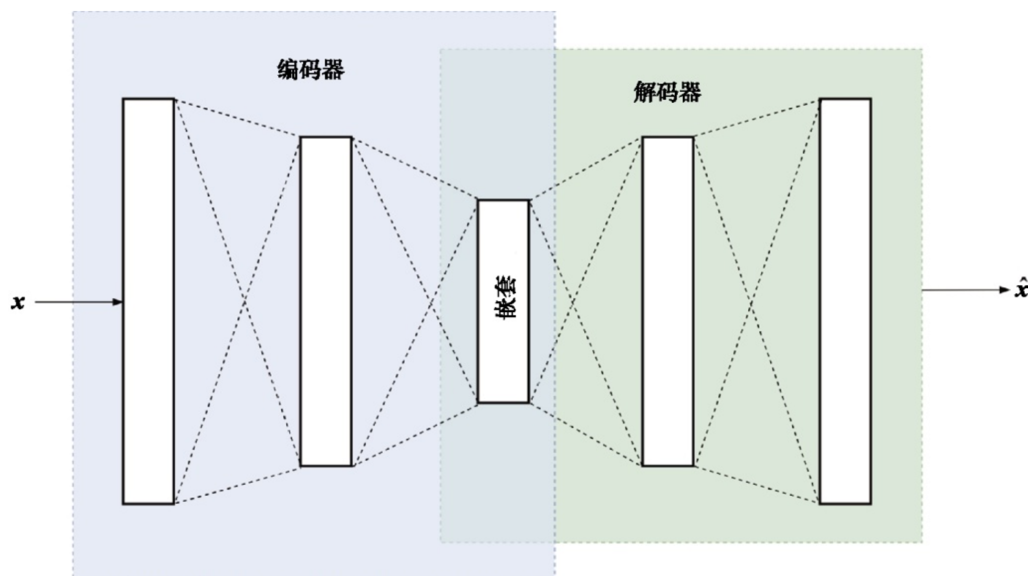
[illegible],[illegible],[illegible],[illegible]SSL [illegible](self-learning)[illegible],[illegible],[illegible]$\boldsymbol{x}$[illegible]([illegible]),[illegible],[illegible],[illegible],[illegible]m[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible]MNIST([illegible],[illegible]0[illegible]9),[illegible]10[illegible]([illegible]100[illegible])[illegible]MNIST[illegible]70 000[illegible](60 000[illegible],10 000[illegible])[illegible](ladder network)[illegible],[illegible]

[illegible]-[illegible],[illegible],[illegible]($\boldsymbol{x}$,$\boldsymbol{x}$)[illegible]$f(\boldsymbol{x})$[illegible]$\hat{x}$[illegible]$\boldsymbol{x}$[illegible]

[illegible],[illegible],[illegible](bottleneck layer)[illegible]D[illegible];[illegible]D[illegible],[illegible]10[illegible]MNIST[illegible]7.6[illegible],[illegible]([illegible])[illegible]([illegible],[illegible]S[illegible])[illegible]



[illegible]7.6 [illegible]

[illegible],[illegible]:

$$\frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i - f(\boldsymbol{x}_i)\|^2$$

[illegible],$\|\boldsymbol{x}_i - f(\boldsymbol{x}_i)\|$[illegible]

[illegible](denoising autoencoder)[illegible]$\boldsymbol{x}$[illegible],[illegible],[illegible]0[illegible]1[illegible],[illegible](normal Gaussian noise)[illegible]$\boldsymbol{x}$[illegible]j,[illegible]$n^{(j)}$[illegible]:

$$n^{(j)} \sim \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(-\mu)^2}{2\sigma^2}\right)$$

[illegible],~[illegible]“[illegible]…[illegible]”,π[illegible]3.141 59…,[illegible]μ[illegible]$\boldsymbol{x}^{(j)}$,[illegible]$\boldsymbol{x}^{(j)}+n^{(j)}$[illegible]

[illegible]([illegible] softmax[illegible])[illegible]l[illegible]l[illegible],[illegible]C_d^l[illegible]([illegible])[illegible],[illegible]C_c[illegible]([illegible])[illegible],[illegible](minibatch stochastic gradient descent with backpropagation)[illegible], $C_c + \sum_{l=1}^{L} \lambda_l C_d^l$ ([illegible])[illegible],[illegible]λ_l[illegible]-[illegible]

[illegible],[illegible]([illegible])[illegible]$\boldsymbol{x}$[illegible],[illegible]

[illegible]

[illegible],[illegible]([illegible]9[illegible])[illegible],[illegible]

[illegible]S3VM,[illegible]SVM[illegible]SVM[illegible],[illegible]S3VM[illegible]

7.10 单样本学习

本节介绍,如何使用单样本学习的方法实现人脸识别。

人脸识别属于单样本学习(one-shot learning),即只需在数据库中存入一张人脸的照片,就可以识别出该人,这与传统的学习算法需要大量的样本训练才能得到一个比较好的模型不同,单样本学习只需要一个训练样例。

对于人脸识别问题,可以利用神经网络进行相似度比较,即学习出一个相似度函数,通过比较两张图片(一张在数据库中)和一张(待识别图片)之间的差异程度,如果差异值小于某个阈值,就可以认为这两张图片是同一个人,否则就不是同一个人。相似度函数需要有这样的性质,相同的人差异值越小,不同的人差异值越大。因此可以通过神经网络来学习这样的相似度函数,这就是孪生神经网络的思想。

一种相似度学习的方法,就是使用孪生神经网络(Siamese Neural Network,SNN)。所谓SNN就是两个相同结构的神经网络,可以是CNN、RNN或MLP等,而且它们共享相同的权重参数;这两个网络的输入分别为两张图片,输出为各自的特征向量,通过比较两个特征向量的距离判断“是同一人”或“不是同一人”,这种方法也常用于签名验证中。

为了训练SNN,可以使用一种三元损失(triplet loss)作为目标函数,训练时需要3张人脸图像作为输入:锚点A(锚点)、正例P(正例)和反例N(反例)的图片,A和P是属于同一人的两张图片;N和它们不是同一个人,由此组成一个三元组(A_i,P_i,N_i)。

假设神经网络对应的函数为f,通过该函数提取每张图片的特征向量,对于训练样本中第i个三元组的损失函数为:

$$\max\left(\|f(A_i)-f(P_i)\|^2-\|f(A_i)-f(N_i)\|^2+\alpha,0\right)$$

(7.3)

□□□□□□□□□□□□□□:

$$\frac{1}{N}\sum_{i=1}^{N}\max\left(\|f\left(A_i\right)-f\left(P_i\right)\|^2-\|f\left(A_i\right)-f\left(N_i\right)\|^2+\alpha,0\right)$$

□□,α□□□□□□□□□□□□,□□□□□A□P□□□□□□,$\|f(A)-f(P)\|^2$□□□□□□;□□□□□□□□□□□□□□□□□,$\|f\left(A_i\right)-f\left(N_i\right)\|^2$□□□□□□□□□□□□□□,$m=\|f\left(A_i\right)-f\left(P_i\right)\|^2-\|f\left(A_i\right)-f\left(N_i\right)\|^2$□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□α□,□□□□m□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□m□□□,□□□□α□□□□,□□□□□□□□□□□□□□□□

□□□□□□□□□□N,□□□□□□□,□□□□□□□□□□□□□□□A□P□□□□□□□N□□□□□□□N□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□

□□□□SNN,□□□□□□□□□□□□□□□□,□□□□□□□□□,□□□□CNN□□□□□□□,□□□□□□□□,□□□□□□□□□□□A□P□N,□□□□7.3□□□□□□□□□□□□□□□□□□□□,□□□□□;□□□□□□□□□□□□□□□□

□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□,□□□□□□□□□□□□□□□□“□□□□□”□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□A□$\hat{A}$□□□□□□□,□□□□□$\left\|f\left(A\right)-f\left(\hat{A}\right)\right\|^2$□□□□□□□□$\tau$□

7.11 □□□□□

□□□□□□□□□□□□□□□□(Zero-Shot Learning,ZSL)□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□



□□,□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□?

□□□□□□□□□□□,□□□□□□□□□□$\boldsymbol{x}$,□□□□□□□$\boldsymbol{y}$□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□:□□□□y_i□y_k□□□□,□□□□□□□□□□□□□□□□□□,y_i□Paris(□□)□y_k□Rome(□□),□□□□□□□□;□□□□,□□y_k□potato(□□),y_i□y_k□□□□□□□□□□□□□□□□□□□□(word embedding),□□□□□□□□□□[1]□□□

□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□4□(□□□□□□,□50□300□),□□4□□□□□□□□□:□□□(animalness)□□□□(abstractness)□□□(sourness)□□□□(yellowness)□□□□□,□□bee(□□)□□□□□[1,0,0,1],□□**yellow**(□□)□□□□□[0,1,0,1],□□□**unicorn**(□□□)□□□□□[1,1,0,0]□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible],[illegible],[illegible]i[illegible]y_i[illegible],[illegible]$\boldsymbol{x}$[illegible],[illegible]f,[illegible]$\hat{y}$[illegible],[illegible]$\hat{y}$[illegible]

[illegible]?[illegible](zebra)[illegible],[illegible](clownfish),[illegible]3[illegible],CNN[illegible](tiger),[illegible],CNN[illegible](mammalness)[illegible](orangeness)[illegible](stripeness)[illegible],[illegible],[illegible],[illegible]

---

[1] [illegible]10[illegible],[illegible]

[illegible]8[illegible] [illegible]

[illegible]“[illegible]”[illegible]

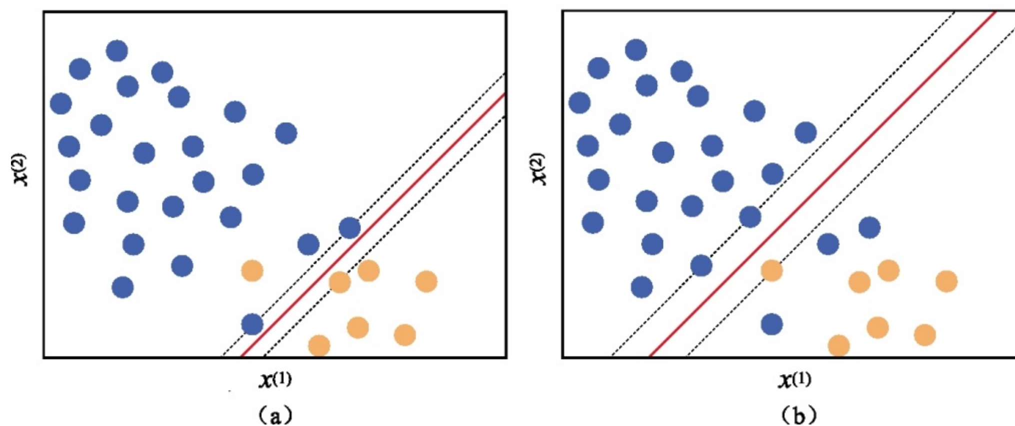
[illegible],[illegible],[illegible]

8.1 [illegible]

[illegible],[illegible],[illegible],[illegible]:[illegible](soft margin)[illegible]SVM,[illegible],[illegible],[illegible]

SVM[illegible],[illegible]“[illegible]”[illegible],[illegible]8.1(a)[illegible]

[illegible],[illegible],[illegible]“[illegible]”[illegible],[illegible]8.1(b)[illegible]



□8.1 □□□□□□□□□

(a)□□□□□□□□□□□;(b)□□□□□□□□□□□□□

□□SVM□□

□□□□□□□□□□□□□□□□,□□□□□□□□□□□□(oversampling)□□□□□□,□□□□□□□□□□,□□□□□□□□□

□□□□□□□□□□□□(undersampling),□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□,□□(synthetic minority oversampling technique,SMOTE)□□□□□□□□□(adaptive synthetic sampling method,ADASYN)□

SMOTE□ADASYN□□□□□□□□□□□□□□□□□□□□$\boldsymbol{x}_i$,□□□□□□□□k□□□(□S_k□□),□□$\boldsymbol{x}_i+\lambda(\boldsymbol{x}_{zi}-\boldsymbol{x}_i)$□□□□□□□□$\boldsymbol{x}_{\mathrm{new}}$□□□,$\boldsymbol{x}_{zi}$□□$S_k$□□□□□□□□□□□□□□□□□□□□□□□$\lambda$□□□0□1□□□□□□□

SMOTE[illegible]ADASYN[illegible]$\boldsymbol{x}_i$[illegible]ADASYN[illegible],[illegible]$\boldsymbol{x}_i$[illegible]S_k[illegible],[illegible],[illegible]

[illegible]

8.2 [illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible]3[illegible]:[illegible](averaging),[illegible](majority vote)[illegible](stacking)[illegible]

[illegible],[illegible],[illegible](base model),[illegible]$\boldsymbol{x}$,[illegible],[illegible]

[illegible]$\boldsymbol{x}$[illegible],[illegible],[illegible],[illegible]([illegible])[illegible]

[illegible]f_1[illegible]f_2,[illegible]$\hat{\boldsymbol{x}}_i=[f_1(\boldsymbol{x}),f_2(\boldsymbol{x})]$[illegible]$\hat{y}_i=y_i$,[illegible]$(\hat{\boldsymbol{x}}_i,y_i)$[illegible]

[illegible],[illegible],[illegible]

□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□

□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□,□□□□□□□:□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□“□□□”□□□□□□□□,□□□□□□□□□□□□□□□□□□□□,□SVM□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□SVM,□□□□□□□□□□

8.3 □□□□□

□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□,□□□□□□□□□□□□□□;□□,□□□□□□□□(standardize),□□□□(normalize)□[0,1]□□□

□□□□□□□□□□□□(□□□□□□,□□□□□□□□□□□□□)□□□□□CNN□RNN,□□□□□□□□□□□(one-hot encoding)□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□(word embedding)□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□(□□□□□□□)□

□□□□□□□□□□□□□□□□□□□□□□□□□□,□seq2seq□□,□□□□□□□□□□□,□□□(Google Scholar)□□□□□(Microsoft Academic)[1]□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□GitHub(□□□□□□)□□□□□□□□□□□□□□,□□□□□□□□

[illegible]

[illegible]

[illegible]

8.4 [illegible]

[illegible] L1 [illegible] L2 [illegible] (dropout) [illegible] (early stopping) [illegible] (batch normalization) [illegible]

[illegible] 0 [illegible] 1 [illegible]

[illegible] 4 [illegible] e [illegible]

神经网络是一种非常灵活的模型,不同的网络可以提供不同层次和视角的信息表达,所以在集成学习中用神经网络作为弱学习器,效果会非常好。另外,在网络训练中,可以定期保存模型的检查点(checkpoint)并在最后将这些保存的模型集成起来;测试时将这几个检查点模型一起集成即可。

此外,还可以采用数据扩增的方式,对同一个图像做各种变换,比如翻转、旋转、平移等,得到多个不同的测试输入,将模型在这些输入上的输出进行平均,同样可以起到提升模型准确率的效果。

当训练数据较少时,神经网络容易发生过拟合,为了缓解过拟合现象,可以使用数据增强(data augmentation)对已有数据进行处理从而产生更多的训练数据,例如对图像进行水平翻转、旋转、裁剪,对音频数据添加噪声等,通过增加训练样本的多样性,提高模型的泛化能力。通过在训练数据上进行各种随机变换,可以使模型不依赖于某些特定的属性,从而提升模型的鲁棒性。

8.5　多模态学习

在生活中,我们接触到的多是多模态(multimodal)的信息数据,如一段视频包含了图像、声音和文本等,这些信息从不同的角度描述了同一事物。

多模态学习就是利用多种模态的数据进行学习,但是这些数据是异构的,在进行学习时需要把它们融合起来,一般有前期融合与后期融合,前者是在特征层面融合,后者是在决策层面融合。

前期融合即在特征层面进行融合,最简单的方法就是将各模态的特征拼接起来(也可以是更复杂的融合方式),例如有图像和文本两种模态的数据,分别得到其特征表示,假设图像特征为$[i^{(1)},i^{(2)},i^{(3)}]$,文本特征为$[t^{(1)},t^{(2)},t^{(3)},t^{(4)}]$,则拼接后的特征为$[i^{(1)},i^{(2)},i^{(3)},t^{(1)},t^{(2)},t^{(3)},t^{(4)}]$。

后期融合即在决策层面进行融合,先对每个模态单独进行训练得到各自的输出结果,然后对各模态的结果进行融合,比如用CNN模型处理图像数据,同时用RNN模型处理文本数据,最后将两个模型的结果进行加权平均或投票。

器;CNN适于提取局部特征,而RNN擅长处理序列特征的信息,用于序列标注和机器翻译等任务,自注意力机制擅长处理长程依赖关系,而softmax和sigmoid分别作用于多分类和二分类的输出,其他激活函数主要用于隐藏层的非线性变换。

8.6 　标签平滑化

在分类任务中,把1作为正样本标签、其他作为负样本标签的独热编码处理方式过于粗暴。实际上,很多样本带有噪声或不同程度地含有多类属性的标签(关键词,tag),这种生硬的处理方式容易导致模型过拟合。

例如,在图像分类或文本分类任务中,样本往往是多义或者带有一定噪声的:一张图片中有多个物体时,其可能既包含了主要类别也包含了次要类别,一段文本也可能与多个主题相关,例如“机器”“人”以及“机器人”这三个类别之间存在一定程度的重叠,如果使用独热编码来处理就会忽略掉这些类别之间的联系。

因此,在分类任务中,需要对样本的标签做一些平滑化处理,例如1这个正样本标签与其他负样本标签之间的差距太大,可以考虑设计一个神经网络模型来学习标签之间的关联性,最简单的方式就是将样本与标签同时输入一个浅层神经网络,激活函数采用ReLU,得到每个类别的语义表示向量,与模型最后一层输出的向量一起计算得到一个表示向量C_1,再将其与样本标签的独热编码向量相乘,得到一个标签的语义表示向量,接着将其作为样本真实标签的语义表示输入softmax,进行归一化,得到一个标签的概率分布向量C_2(平滑化后的标签)。

得到平滑化后的标签后就可以将其与原始标签结合起来,作为新的样本标签,参与模型的训练。其中一种简单的结合方式就是加权求和,比如对原始标签向量乘以一个(0,1)之间的数γ,那么新的标签向量就是$\gamma C_i + (1-\gamma) C_2$。在实际的应用场景中,可以通过网格搜索来找到最优的γ取值。

8.7 　文本增强

在深度学习中,通过一种被称为迁移学习(transfer learning)的方法可以将在一个场景中学到的知识,用在另一个场景中。但是在常识推理中,把学到的技能迁移到一个全新的环境中是一个挑战。在生活中,人总是能够轻松地把一个场景中学习到的东西迁移到另一个场景,这对于现阶段的机器来说很困难,而这正是常识的一部分。

人工智能要实现常识推理需要把大量的常识知识以一种计算机(机器人)可用的方式表示,并采用适当的机制,解决两个关键问题:一是如何把常识或者世界知识赋予计算机或者机器人,二是如何进行常识推理:如何让机器像人一样思考。

就人类常识推理而言,常识的范围非常广,其中最常用的常识包括如下方面:

(1)对于物体的常识和日常(物理常识)物体相互作用的理解。

(2)对人类行为(心理常识)及人和人之间交流的理解。

(3)对世界上时间和空间上一连串事件的理解,包括各种事物的结构及事物之间相互影响的机制。

(4)对世界的认识,对具体的事物知道它的意义。

(5)知道如何根据事物的环境进行“变通”。

(6)对具体事物,可以获得它的意义,以及可能的结果及影响。

目前,尽管我们已经有了学习数学和逻辑推理等极为抽象的思维规则的计算机,但是常识推理背后的机制(人类知识是如何获得并表示的),仍然是人工智能最难攻克的难题。

人类的常识是非常复杂的,而且很难给出严格的定义,这就使得在计算机中建立常识知识库变得非常困难。例如,我们让机器人在某个规定的时间(比如午饭时间)给主人倒一杯咖啡,如果1 000

[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible]“[illegible]”[illegible],[illegible],[illegible]

8.8 [illegible]

[illegible],[illegible],[illegible]

[illegible](analysis of algorithm)[illegible],[illegible]**O**[illegible](Big O notation)[illegible],[illegible],[illegible]

[illegible],[illegible]1[illegible]N[illegible]S[illegible]([illegible]Python[illegible],[illegible]):

```
1  def find_max_distance(S):
2      result = None
3      max_distance = 0
4      for x1 in S:
5          for x2 in S:
6              if abs(x1 - x2) >= max_distance:
7                  max_distance = abs(x1 - x2)
8                  result = (x1, x2)
9      return result
```

[illegible],[illegible]S[illegible],[illegible]S[illegible],[illegible]N^2[illegible]comparison([illegible])[illegible]abs([illegible])[illegible]assignment([illegible])[illegible],[illegible]([illegible])[illegible]

$5N^2$(□□□□□1□comparison□2□abs□2□assignment□□□□□□□□□□□□□□□□□□□□□,□□□□O□□□□□□□□□□□□$O(N^2)$□□□,□□□□5,□□□□

□□□□□,□□□□□□□□□□:

```python
def find_max_distance(S):
    result = None
    min_x = float("inf")
    max_x = float("-inf")
    for x in S:
        if x < min_x:
            min_x = x
          if x > max_x:
            max_x = x
    result = (max_x, min_x)
    return result
```

□□□□□□□,□□□□□□□□S□□□□□,□□□□□□□□$O(N)$□□□,□□□□□□□□□□□□□□□□(more efficient)□

□□□□□□□□□□□□□□□(polynomial)□,□□□□□□□□□,$O(N^2)$□$O(N)$□□□□□,□□N□N^2□□□N□1□□□□2□□□□□□,□□□□□□,□□$O(N^2)$□□□□□□□□□□□□□□□,□□□□□□□□$O(\log N)$□□□

□□□□□□,□□□□□,□□□□□□□□□□(avoid using loops whenever possible),□□□□□□□□□□□□□□□Python□□□,□□□wx,□□□:

```python
import numpy
wx = numpy.dot(w,x)
```

□□□:

```
wx = 0
for i in range(N):
    wx += w[i]*x[i]
```

□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□(set)□□□□□(list)□□□□□□Python□,□□□□□□□□□□□*x*□□□*S*□□□,□□□*S*□□□□□□□□□□,□□*S*□□□□□□□□□□□

□□□□□□Python□□□□□□□□□□□□□□(dict)□□□□□□□□□□□□□□□□(hashmap)□□□□□□□□□□□□□□(key-value pair)□□□,□□□□□□□□□□□□□

□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Python□□numpy□scipy□□scikit-learn,□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□(method)□C□□□□□

□□□□□□□□□□□□□□□□□□□□□,□□□□□(generator)□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□

□□□□□□□Python□□cProfile□□□□□□□□□□□□□□□

□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□:

- □□multiprocessing□□□□□□□□
- □□PyPy□Numba□□□□□□Python□□□□□□□□□□□□□□□□□□

---

[1] □□□:□□□□□□□□□□□□□□□□□

第9章 核密度估计

在本书中多次讨论过核方法在各种数据挖掘任务中的成功应用。本章将介绍核方法的另一个重要应用——核密度估计。通过本章的学习,读者将掌握核密度估计的基本原理和带宽选择方法,并能够运用核密度估计解决实际问题中的概率密度估计问题,对核方法有更深入的理解。

9.1 密度估计

密度估计(density estimation)是这样一个问题:给定一组从某个未知分布中抽取的样本,如何估计该分布的概率密度函数(probability density function,pdf)。密度估计在许多领域有广泛的应用,例如在本书前面的章节中的第7章中,我们使用了估计的pdf来构建分类器。在一些情况下,可以采用参数化(parametric)方法,即假设数据服从某种已知的分布(如多元正态分布(multivariate normal distribution,MVN)),然后利用数据估计分布的参数。但是在许多情况下,数据并不服从MVN分布,或者我们对数据的分布一无所知,此时,无法做假设,就需要使用非参数(nonparametric)的估计方法,核密度估计就是其中之一。

设$\{x_i\}_{i=1}^{N}$是一组独立同分布(从某个分布抽样)的样本,其所服从分布的pdf记为f,其中对i=1,…,N,每个$x_i \in \mathbb{R}$是标量,我们希望从f的样本中估计f。核密度$\hat{f}_b$定义如下:

$$\hat{f}_b(x) = \frac{1}{Nb}\sum_{i=1}^{N} k\left(\frac{x - x_i}{b}\right) \tag{9.1}$$

其中,b是带宽或者平滑参数的正实数,k是核函数。第7章中说,高斯核函数是这样的:

$$k(z) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-z^2}{2}\right)$$

通过选择带宽b值,使真实f和核密度估计量$\hat{f}_b$之间的误差最小化。这个误差可用平均积分平方误差(mean integrated squared error,MISE)度量:

$$\mathrm{MISE}(b) = \mathbb{E}\left[\int_{\mathbb{R}} \left(\hat{f}_b(x) - f(x)\right)^2 \mathrm{d}x\right]$$

(9.2)

在式9.2中,积分是真实pdf f和估计$\hat{f}_b$之间的差值的平方的积分$\int_{\mathbb{R}}$。它是数值的近似。实际上,它可能是$\sum_{i=1}^{N}$的形式,而期望值运算符$\mathbb{E}$的近似形式可能是$\frac{1}{N}$。

为了更好地理解这个公式,对$(\hat{f}_b(x) - f(x))^2$,在每个点上计算差值的平方。期望值运算符$\mathbb{E}$保证了对于带宽b的选择不依赖于$\{x_i\}_{i=1}^{N}$中一组具体的样本或观察值,同时$\hat{f}_b$在所有可能的样本中平均表现良好,而真实的pdf f的积分被整个域($\mathbb{R}$定义)所计算。

可以展开式9.2的平方部分得到:

$$\mathbb{E}\left[\int_{\mathbb{R}} \hat{f}_b^2(x)\,\mathrm{d}x\right] - 2\mathbb{E}\left[\int_{\mathbb{R}} \hat{f}_b(x)\, f(x)\,\mathrm{d}x\right] + \mathbb{E}\left[\int_{\mathbb{R}} f(x)^2 \mathrm{d}x\right]$$

最后一项不依赖于我们所选的b,因此可以忽略它。第一项很容易计算$\int_{\mathbb{R}} \hat{f}_b^2(x)\,\mathrm{d}x$;中间项可以通过一个称为交叉验证(cross-validation)的近似$-\frac{2}{N}\sum_{i=1}^{N} \hat{f}_b^{(1)}(x_i)$,其中$\hat{f}_b^{(1)}$是$f$的核模型,不包括第$x_i$个观察值的估计。

$\sum_{i=1}^{N} \hat{f}_b^{(1)}(x_i)$□□□□□□□□□□□□□(leave one out estimate)□□□□□□□□□□□,□□□□(fold)□□□□□□□□□□□□□□□□,$\int_{\mathbb{R}} \hat{f}_b(x) f(x)\,\mathrm{d}x$□(□□□□□$a$)□□□$\hat{f}_b$□□□□,□□$f$□□□pdf□□□□□□□□□□□□$\mathbb{E}[a]$□□□□□□

□□□,□b□□□□b^*□□□□□□□□□□□:

$$\int_{\mathbb{R}} \hat{f}_b^2(x)\,\mathrm{d}x - \frac{2}{N} \sum_{i=1}^{N} \hat{f}_b^{(1)}(x_i)$$

□□,□□□□□□□□□□□□□□D□□□□□$\boldsymbol{x}$,□9.1□□□□□x-x_i□□□□□□□□||$\boldsymbol{x}$-$\boldsymbol{x}_i$||□□□9.1□,□□□□100□□□□3□□□□b□□□□□pdf□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□b^*□

（a）拟合良好

（b）过拟合

（c）欠拟合

（d）网格搜索

[illegible]9.1 [illegible]

9.2 [illegible]

[illegible](clustering)[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible]

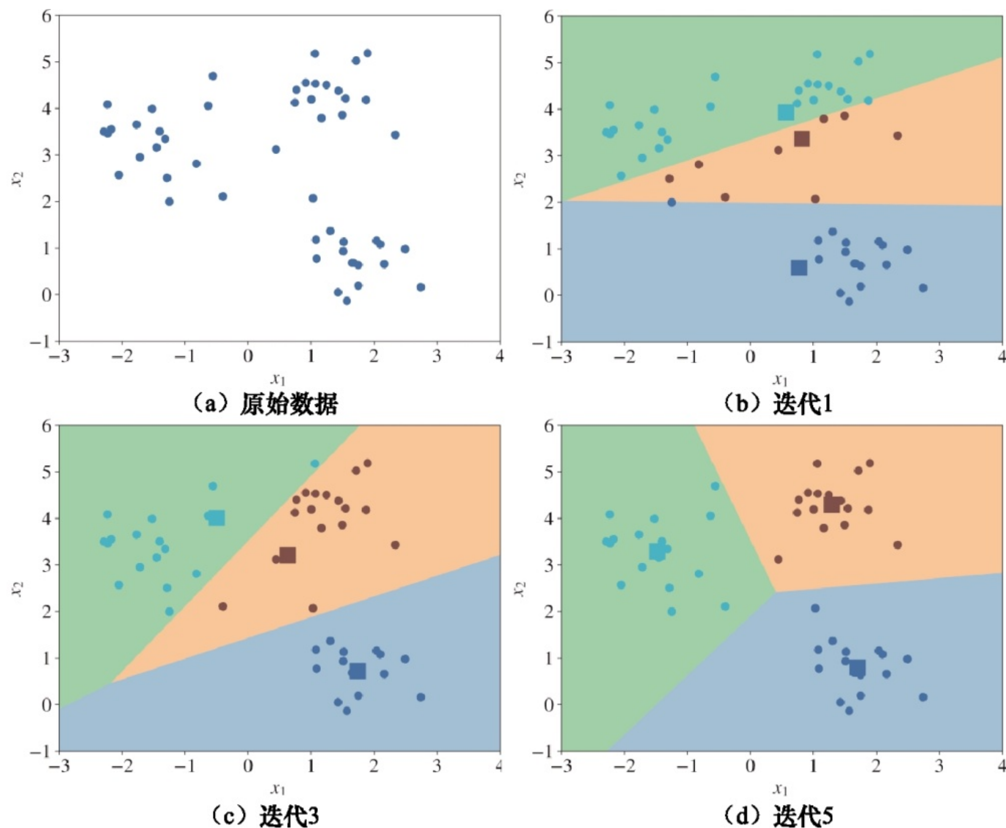
9.2.1 k[illegible]

k[illegible](k-mean)[illegible],[illegible]*k*[illegible](cluster)[illegible],[illegible]*k*[illegible],[illegible](centroid)[illegible],[illegible],[illegible],[illegible]***x***[illegible]c[illegible],[illegible]([illegible])[illegible],[illegible]

[illegible],[illegible],[illegible]

[illegible],[illegible]k[illegible]k[illegible],[illegible]

[illegible]k[illegible]9.2[illegible]9.2[illegible],[illegible];[illegible]



□9.2 □k=3□,k□□□□□□□

□□□k□□□□□□□□□,□□□□□□□□□□□□□□□□k□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□”,□□□□□□□□□□□□□□□□□,□□□□□□□□□k□□□□□□□□□□□□□□□□□□□□□□□□□

9.2.2 DBSCAN□HDBSCAN

DBSCAN□□□□□□□□□□,□□□k□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□:ϵ□n□□□□□,□□□□□□□□□□□□□□□$\boldsymbol{x}$,□□□□□□□1□□□,□□□□□□□□$\boldsymbol{x}$□□□□□□□□□□ϵ□□□□□□□□□□n,□□□□ϵ□□□□□□1□□□□□□□□□1□□□□□□,□□□□□ϵ□□□□□□□□□n□□□□ϵ□□,□□ϵ□□□□□□□□1□□□□□□

为1的正方形内部的点随机生成的均匀数据,我们可以看到它的大部分点都被归到了一个簇中,而在一个边长为2的正方形中,点的密度相对比较低的情况下,就变得稀疏起来。所以我们在选择邻域半径ϵ和最小邻居数n时要很小心,并且最好是多尝试几次。

DBSCAN算法的优点,除了前面提到的可以识别出非凸数据的簇之外,k均值算法对离群点非常敏感。这个缺点在它能够分开离群点的DBSCAN算法身上就不再存在了。不过,它对邻域半径的选择(也就是ϵ)会非常敏感,而且对ϵ较大,则适用于没有显著密度差异的数据集。

HDBSCAN算法是基于DBSCAN的改进,它不需要指定ϵ参数而是使用层次化的思想去构建聚类,能够处理不同密度的簇,具有更稳健的性质。

HDBSCAN只需要最小邻居数n,这一个参数而且比较容易确定,因此比较适合实际应用。在使用HDBSCAN时比较推荐的做法是:先用一个小的参数去检查数据的大致结构,然后用k均值算法或者HDBSCAN再聚类,也就是将它们结合起来使用。由于其中的很多内容需要较多篇幅介绍,有兴趣的读者请参考HDBSCAN。

9.2.3 聚类数量的选取

在实际工作中,我们可能会面对一个难题就是确定聚类的数量。在某些情况下,我们可能对所要聚类的数据已经有一定的了解从而能够决定要分成几个簇,但在多数情况下,我们需要借助一些方法[1]来帮助我们进行判断。

预测强度方法预测强度(prediction strength)可以用来选择聚类的数量。它的思路是这样的,首先把数据集分成两个部分,一个作为训练集,一个作为测试集,例如训练集有N_{tr}个样本即S_{tr};测试集有N_{te}个样本即S_{te}。选定k个簇,对S_{tr}和S_{te}分别使用聚类算法C得到聚类结果为$C(S_{tr},k)$和$C(S_{te},k)$。

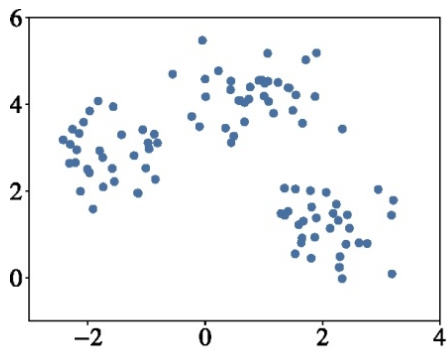
□A□□□□□□□□□□□$C(S_{tr},k)$□A□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□k□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□k□□□□□□,□□9.2□□□

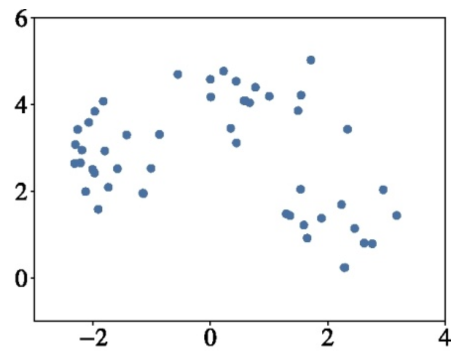
□□,□□□□□□□$N_{te}\times N_{te}$□□□□□□□(co-membership matrix)$\boldsymbol{D}[A,S_{te}]$:□□□□□□□□□□□□□$\boldsymbol{x}_i$□$\boldsymbol{x}_{i'}$□A□□□□□□□□□□,$\boldsymbol{D}[A,S_{te}]^{(i,i')}=1$;□□$\boldsymbol{D}[A,S_{te}]^{(i,i')}=0$□

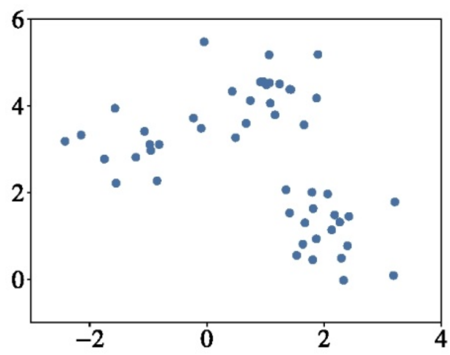
□□,□□□□□□□□□□□□□□□□□□□□□□□□□□k□□□□□□□□A□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□A□

□□□□□□□□□□,□□k□□□□□□□□□□□,□□□□□$C(S_{te},k)$□□□□□□□□□□□□,□□□$C(S_{tr},k)$□□□□□□□□□□□□□□□□,□□k□□□□□□(□□□□□),□□□□□□□□□□□□□□□□□□□□□

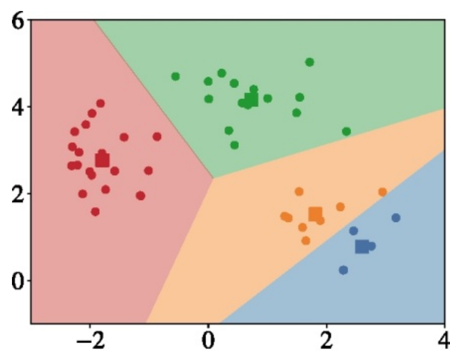
□9.3□□9.4□□□□□□□□□□□□□□□9.4(a)□□9.4(b)□□□□□□$C(S_{tr},k)$□□□$C(S_{te},k)$□□□□□□□9.4(c)□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□$\boldsymbol{D}[A,S_{te}]$□□□□□0□,□□k=4□□□□□□□



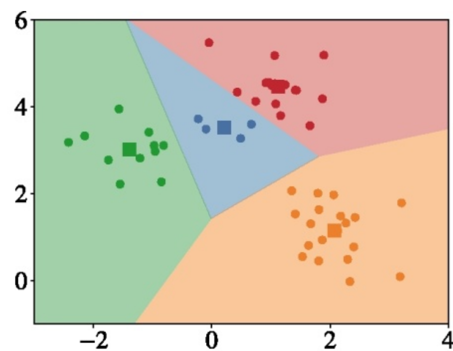




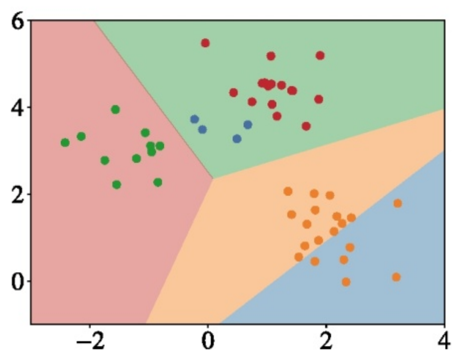
[illegible]9.3 [illegible]9.4[illegible]



（a）训练数据聚类方案



（b）测试数据聚类方案



（c）测试数据重叠于训练数据的聚类方案

[illegible]9.4 [illegible]k=4[illegible]

[illegible],k[illegible]:

$$\mathrm{ps}\,(k) \stackrel{\mathrm{def}}{=} \min_{j=1,\ldots,k} \frac{1}{|A_j|\,(|A_j|-1)} \sum_{i,i' \in A_j} \boldsymbol{D}[A, S_{te}]^{(i,i')}$$

[illegible],$A \stackrel{\mathrm{def}}{=} C(S_{tr}, k)$[illegible]$C(S_{tr},k)$[illegible]$j$[illegible],$|A_j|$[illegible]$A_j$[illegible]

[illegible]$C(S_{tr},k)$,[illegible],[illegible]k[illegible]

□□□□□□,□□□□□□□□□□□□ps(k)□□0.8□□□k□□□k□□□2□□3□□4□□□□□□□□□□□□9.5□□□

图9.5　□□k□□□2□□3□□4□□□□□□□□□□

□□□□□□□□,□k□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□k□,□□□□□□□□□□□□□□$\overline{\mathrm{ps}}(k)$□

□□□□□□□□□□□□□□□□□□□□(gap statistic)□□□□□□□□□□□□□(elbow method)□□□□□□(average silhouette method)□□□□□□□□□□□□

9.2.4 □□□□□□

DBSCAN□k□□□□□□□□(hard clustering),□□□□□□□□□□□□□□□□□□□(Gaussian mixture model,GMM)□□□□□□□□□□□□□,□□□□□□□□□□(membership score)(HDBSCAN□□□)□□□GMM□□□□□□□□□□□□□□□□□□□GMM□,□□□□□□□□□□□(MND),□□□□□□□MND□□□□□:

$$f_X = \sum_{j=1}^{k} \phi_j f_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}$$

□□,$f_{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j}$□□□MND j,□ϕ_j□j□□□□□□□□□□□□□$j=1,\ldots,k$,□□$\boldsymbol{\mu}_j$,$\boldsymbol{\Sigma}_j$□ϕ_j□□□□□□□□□□□(expectation maximization algorithm,EM)□□□□□□□(maximum likelihood)□□□

□□□□□□,□□□□□□□□□□□□□□□□□:k=2□□□,□□□□□□□□□:

$$f(x|\mu,\sigma_1^2) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp - \frac{(x-\mu_1)^2}{2\sigma_1^2},$$
$$f(x|\mu,\sigma_2^2) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp - \frac{(x-\mu_2)^2}{2\sigma_2^2}$$

(9.3)

□□,$f(x|\mu_1,\sigma_1^2)$□$f(x|\mu_2,\sigma_2^2)$□□□pdf,□□$X=x$□□□□□

□□□□EM□□□□$\mu_1,\sigma_1^2,\mu_2,\sigma_2^2,\phi_1\phi_2$□□□□□$\phi_1$□$\phi_2$□□□□□□□□□□□□□□□□□□,□□□□□□□□

EM□□□□□□□□□□□□,□□□□μ_1□σ_1^2□μ_2□σ_2^2□□□□,□□$\phi_1=\phi_2=\frac{1}{2}$[□□□□□,□□□□□$\phi_j$($j$=1,…,$k$)□□$\frac{1}{k}$]□

□EM□□□□□□,□□□□4□□□:

(1)□□□i=1,…,N,□□□9.3□□□□x_i□□□□

$$f(x_i|\mu_1,\sigma_1^2) \leftarrow \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp - \frac{(x_i-\mu_1)^2}{2\sigma_1^2},$$

$$f(x_i|\mu_2,\sigma_2^2) \leftarrow \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp - \frac{(x_i-\mu_2)^2}{2\sigma_2^2}$$

(2)□□□□□□□(Bayes’ Rule),□□□□□x_i,□□□□□□□□j∈{1,2}□□□□$b_i^{(j)}$(□□□□,□□□□□□□□□j□□□□):

$$b_i^{(j)} \leftarrow \frac{f(x_i|\mu_j,\sigma_j^2)\phi_j}{f(x_i|\mu_1,\sigma_1^2)\phi_1 + f(x_i|\mu_2,\sigma_2^2)\phi_2}$$

□□ϕ_j□□□□,□□□□□□μ_j□σ_j^2□□□□□j□□□□□□□□□□□□□□□□□□□□□□□□$\phi_1=\phi_2=\frac{1}{2}$□□□:□□□□□□□□□□□□,□□□□□□□□□□□□□□

(3)□□□□μ_j,σ_j^2,j∈{1,2}:

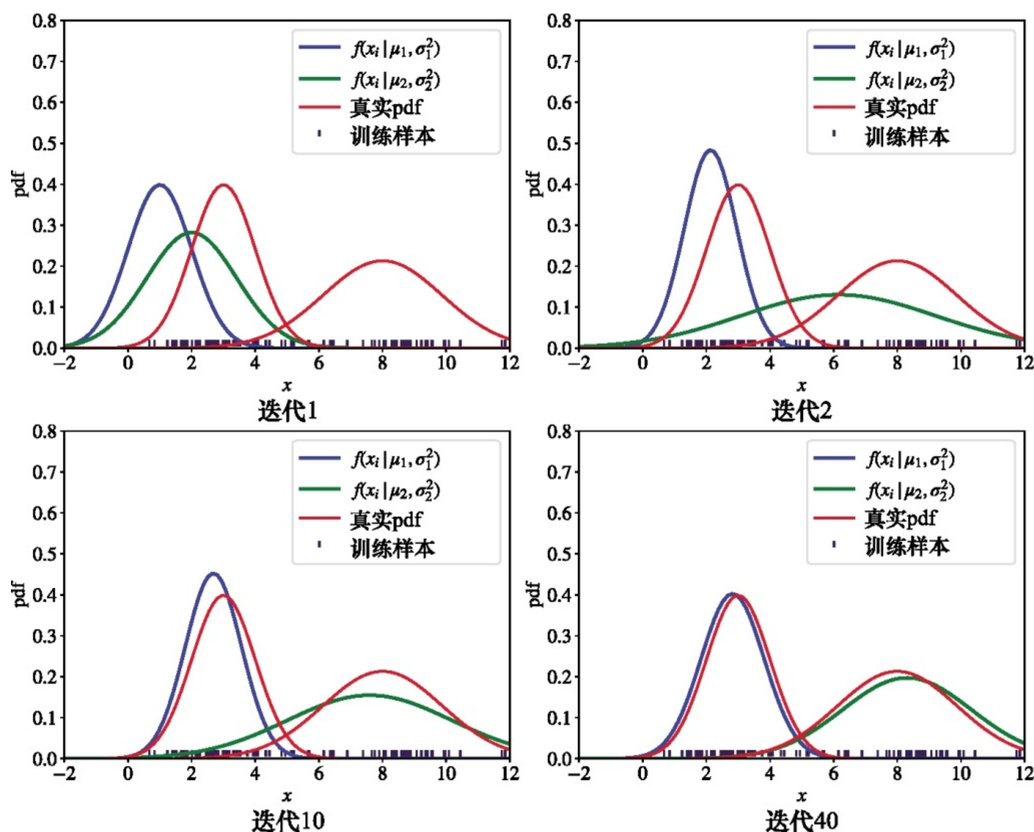
$$\mu_j \leftarrow \frac{\sum_{i=1}^{N} b_i^{(j)} x_i}{\sum_{i=1}^{N} b_i^{(j)}} \quad □ \quad \sigma_j^2 \leftarrow \frac{\sum_{i=1}^{N} b_i^{(j)} (x_i-\mu_j)^2}{\sum_{i=1}^{N} b_i^{(j)}}$$

(9.4)

(4)□□$\phi_j, j \in \{1,2\}$

$$\phi_j \leftarrow \frac{1}{N}\sum_{i=1}^{N} b_i^{(j)}$$

□□□□□□□(1)□(4),□□μ_j□σ_j^2□□□□□□□□□:□□□□□□□□□□ϵ□□□□□□□9.6□□□



□9.6 □□EM□□□□□□□□(k=2)□□□□□□□□□□□□

□□□□□□□EM□□□k□□□□□□□□:□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□,□□GMM□□□□□□x_i□□□j□□□□(soft clustering):x_i□$b_i^{(j)}$□□□□□□□j□□□□□□,□□□□□□9.4□□μ_j□σ_j^2□□□□□□□□□□(□k□□),□□□□□□$b_i^{(j)}$□□□□□□

□□□□□□j□□□μ_j□σ_j^2□□,□□x□□□j□□□□□□□□$f(x_i|\mu_j,\sigma_j^2)$□□□□□

□□□□□□□□□□D□□□(D>1)□□□□□□□σ^2□□□□□□□Σ□□□□□□□□□□□

□□□□□□□(MND)□



k□□□□□□□□□□□□□,□GMM□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□

□GMM□,□□□□□□□□□□□□k□□□□□□□□□□□□□□□□□□□□□□□□□□□

□,□□□□□□□□□□□k□□□□□□□f_{tr}^k□□□,□□□□□□□□□□□□□□□□□k:

$$\arg\max_k \prod_{i=1}^{|N_{te}|} f_{tr}^k(\boldsymbol{x}_i)$$

□□,$|N_{te}|$□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□(spectral clustering)□□□□□(hierarchical clustering)□□□□□□□□□□□□□□□□□□,□□□□□□□□□,k□□□HDBSCAN□□□□□□□□□□□□□□□□

9.3 □□□□

随着数据量的增大,同时也得益于硬件的发展,本部分所讲解的降维方法主要针对中等规模的数据集。使用图形处理器(GPU),有望将这些方法进一步扩展到更大规模的数据集。具体的应用场景包括:数据降维、数据可视化等。

一般情况下,降维的目的是在保持数据之间关系的前提下,将高维数据映射到低维空间中,从而使得数据更加易于理解和处理。在实际应用中,降维后的数据可以作为其他机器学习算法的输入,或者直接用于数据可视化。数据降维的方法有很多种,本节将介绍其中的几种——线性降维和非线性降维方法。

本节主要介绍的降维方法有3种,即主成分分析法(Principle Component Analysis,PCA)、均匀流形近似与投影(Uniform Manifold Approximation and Projection,UMAP)和自动编码器(autoencoder)。

读者在第7章中已经学习了聚类的相关内容。聚类和降维都属于无监督学习,它们之间有着紧密的联系。在实际应用中,可以先对数据进行降维,然后再对降维后的数据进行聚类分析,以获得更好的效果。

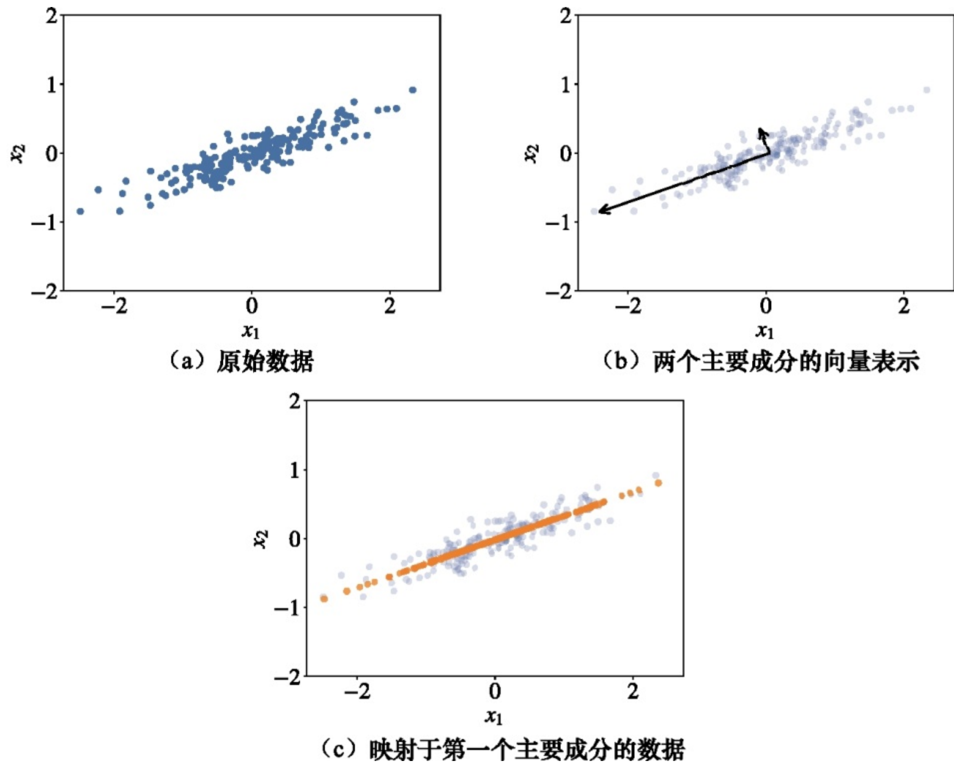
9.3.1 主成分分析法

主成分分析(PCA)是一种常用的线性降维方法。它的基本思想是将原始数据投影到2个或多个相互正交的方向上,使得投影后的数据方差最大化。这些方向被称为主成分,它们是原始特征的线性组合,能够最大程度地保留原始数据的信息。

以二维数据为例,如图9.7(a)所示,数据点分布在一个椭圆形区域内。主成分分析会首先找到数据方差最大的方向作为第1主成分,然后找到与第1主成分正交且方差次大的方向作为第2主成分。第1主成分方向上的方差最大,而在第2主成分方向上的方差较小。通过只保留第1主成分,

□3□□□□□□□□□□□□,□□□□□□3□□□□,□□□□□□□9.7(b)□,□□□□□□□□□□□

□□□,□□□□□□□□□□□□□□



□9.7　PCA



□□,□□□□□□□□□□□□□□$D_{new}<D$,□□□□D_{new}□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□$D_{new}=1$,□□□□□□□□□□□□□□□□□□□9.7(c)□□□□□□

除了它们的知名度外,如果我们要可视化数据,就可以使用它们:首先我们需要训练一个模型来对数据进行分类或回归,然后选择模型的一个全连接层,取所有数据的2维或3维嵌入向量来作为嵌入层的输入。它将转化为2D或3D向量,并通过可视化其分类属性来表示数据。

9.3.2 UMAP

在实践中我们观察到,目前许多最流行的降维算法都是基于相似性度量的,如t-SNE和UMAP,它们都是基于数据点的相似性进行降维。相似性度量这一问题定义是无监督的,也就是说没有目标函数的,不同的定义都会有不同的结果,它们各有各的优点和缺点。

在UMAP中,相似性度量函数w定义为:

$$w\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \stackrel{\text{def}}{=} w_i\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)+w_j\left(\boldsymbol{x}_j, \boldsymbol{x}_i\right)-w_i\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) w_j\left(\boldsymbol{x}_j, \boldsymbol{x}_i\right)$$

(9.5)

其中$w_i(\boldsymbol{x}_i,\boldsymbol{x}_j)$定义为:

$$w_i\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \stackrel{\text{def}}{=} \exp \left(-\frac{d\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)-\rho_i}{\sigma_i}\right)$$

其中,$d(\boldsymbol{x}_i,\boldsymbol{x}_j)$是两个样本之间的欧几里得距离,$\rho_i$是$\boldsymbol{x}_i$到最近邻样本的距离,而$\sigma_i$是$\boldsymbol{x}_i$到它的$k$个最近邻的距离($k$是一个超参数)。

公式9.5的度量值是对称的,它的取值范围介于0到1,并且满足对称性条件,$w(\boldsymbol{x}_i,\boldsymbol{x}_j)=w(\boldsymbol{x}_j,\boldsymbol{x}_i)$。

设w表示输入数据的原始空间中的相似度;设w'表示降维后的嵌入（即低维）空间中的相似度,如式9.5所示。

在模糊集理论中,对于给定的一个模糊集合(fuzzy set)而言,每一个元素都有隶属度。假设集合S中的任一元素x都可以表示成数值$\mu_S(x) \in [0,1]$,表示x属于S的隶属程度(membership strength)。如果$\mu_S(x)$接近0,则说明元素x几乎不属于模糊集S的元素,如果$\mu_S(x)$接近1,则x几乎完全属于集合S。如果对于$x \in S$都满足$\mu(x)=1$,则集合S就变成了我们熟知的普通集合或者经典集合。

用以上的模糊集的概念来解释这里的相似度。

因为w和w'都是位于0到1之间,所以可以用认为$w(\boldsymbol{x}_i,\boldsymbol{x}_j)$是表示数据对$(\boldsymbol{x}_i,\boldsymbol{x}_j)$属于一个模糊集合的隶属度。$w'$亦如此。于是就可以用交叉熵来度量两个模糊集合之间的差异(fuzzy set cross-entropy),其定义:

$$C_{w,w'} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left[w\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)\ln\left(\frac{w\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)}{w'\left(\boldsymbol{x}'_i,\boldsymbol{x}'_j\right)}\right) + \left(1-w\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)\right)\ln\left(\frac{1-w\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)}{1-w'\left(\boldsymbol{x}'_i,\boldsymbol{x}'_j\right)}\right)\right]$$

(9.6)

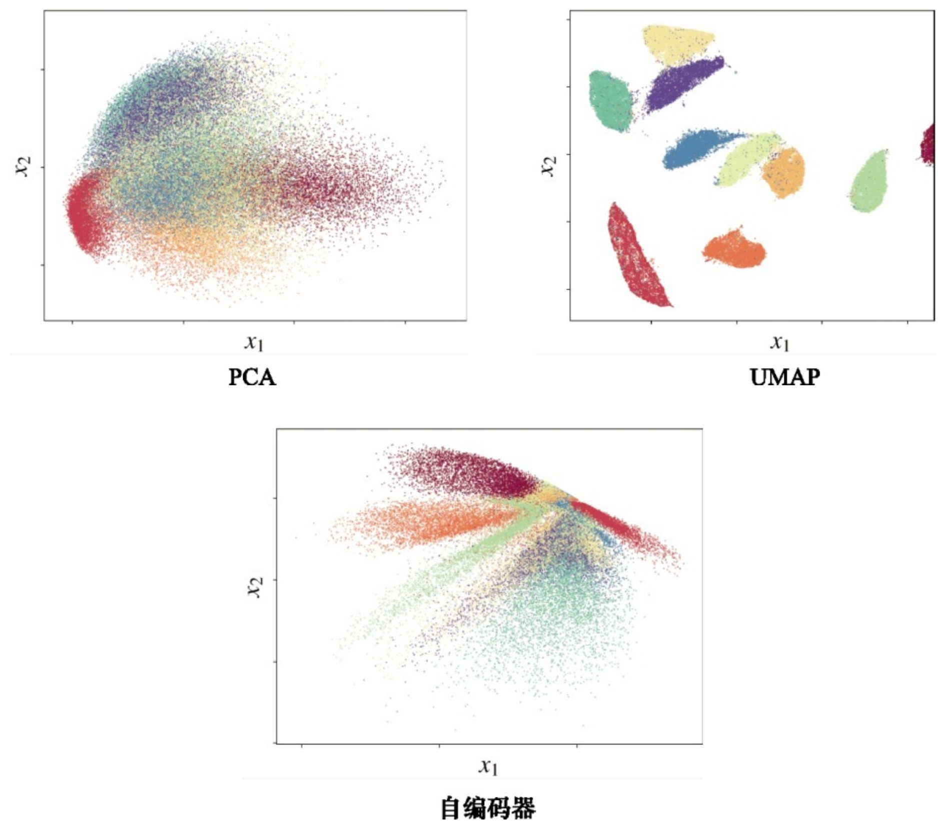
其中,$\boldsymbol{x}'$是在低维空间中$\boldsymbol{x}$所对应的“映射”。

对于9.6式,求解最优的$\boldsymbol{x}'_i$(其中，i=1,…,N),就是降维的结果，通常使用随机梯度下降的方法，求解使得$C_{w,w'}$最小。

如图9.8所示,是使用经典的MNIST数据集降维之后的可视化效果。关于MNIST数据集，它是一个手写数字数据集:共包含70 000个样本，在图9.8中的10种颜色表示对应10个数字,有的颜色的

[illegible],UMAP[illegible]([illegible])[illegible]

[illegible],UMAP[illegible]PCA[illegible],[illegible]



[illegible]9.8 [illegible]MNIST[illegible]3[illegible]

9.4 [illegible]

[illegible](outlier detection)[illegible]

[illegible]:[illegible]

[illegible],[illegible],[illegible]

[illegible],[illegible]

[illegible],[illegible],[illegible]

[1] [illegible]

[illegible]10[illegible] [illegible]

10.1 [illegible]

[illegible],[illegible]([illegible])[illegible](Euclidean distance)[illegible](cosine similarity)[illegible],[illegible]([illegible])[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible]k[illegible]kNN[illegible],[illegible]?[illegible],[illegible]

[illegible]$\boldsymbol{x}$[illegible]$\boldsymbol{x}'$[illegible]:

$$d\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \left\|\boldsymbol{x} - \boldsymbol{x}'\right\| \stackrel{\text{def}}{=} \sqrt{\left(\boldsymbol{x} - \boldsymbol{x}'\right)^2} = \sqrt{\left(\boldsymbol{x} - \boldsymbol{x}'\right)\left(\boldsymbol{x} - \boldsymbol{x}'\right)}$$

[illegible],[illegible],[illegible]:

$$d_{\boldsymbol{A}}\left(\boldsymbol{x}, \boldsymbol{x}'\right) = \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_{\boldsymbol{A}} \stackrel{\text{def}}{=} \sqrt{\left(\boldsymbol{x} - \boldsymbol{x}'\right)^{\mathrm{T}} \boldsymbol{A} \left(\boldsymbol{x} - \boldsymbol{x}'\right)}$$

[illegible],$\boldsymbol{A}$[illegible]$D\times D$[illegible]D=3,[illegible]$\boldsymbol{A}$[illegible]:

$$\boldsymbol{A} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

□d_A□□□□□□□□□□□□□□□□□□□□:

$$\boldsymbol{A} \stackrel{\text{def}}{=} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

□□□□□□□□□□□□(□□□□,□□□□□□□□□□□□)□□□,□□□□□□□□□□□□□□□□3□□□:

- $d(\boldsymbol{x},\boldsymbol{x}')\geq 0$ □□□
- $d(\boldsymbol{x},\boldsymbol{x}')\leq d(\boldsymbol{x},\boldsymbol{x}')+d(\boldsymbol{x}',\boldsymbol{z})$ □□□□□
- $d(\boldsymbol{x},\boldsymbol{x}')=d(\boldsymbol{x}',\boldsymbol{x})$ □□□

□□□□□□□□,$\boldsymbol{A}$□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□$\boldsymbol{M}$□□□:

$$\boldsymbol{z}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{z} \geqslant 0$$

□□,$\boldsymbol{z}$□□$\boldsymbol{M}$□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□

□□□□□□□□,□□□□□□□□$(d(\boldsymbol{x},\boldsymbol{x}')+d(\boldsymbol{x}',\boldsymbol{x}))/2$□

□□□□□□□□□□□□$X = \{\boldsymbol{x}_i\}_{i=1}^{N}$□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□S□□□□□□□□□□□□$(\boldsymbol{x}_i,\boldsymbol{x}_k)$;□□□□□$D$□□□□□□□□$(\boldsymbol{x}_i,\boldsymbol{x}_k)$,□□□□□□□

□□□□□□□□□□$\boldsymbol{A}$,□□□□□□□□□□□□□:

$$\min_{\boldsymbol{A}} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_k)\in S} \|\boldsymbol{x}-\boldsymbol{x}'\|_{\boldsymbol{A}}^{2} \quad \text{□□} \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_k)\in D} \|\boldsymbol{x}-\boldsymbol{x}'\|_{\boldsymbol{A}} \geqslant c$$

其中,$\boldsymbol{A}$是一个半正定阵,c是一个边界超参数(一般为正数)。



通过上述损失函数,可以得到一个有效的度量,从而学到半正定阵$\boldsymbol{A}$。关于度量学习更多的内容请扫码。

在深度学习中,一次性学习(one-shot learning)中的孪生网(Siamese network)、三元组损失(triplet loss)等均是度量学习的思想:集合S中的人脸图像在测试时是同一个人;而集合D中的人脸图像则不同。

度量学习在第一章已有所提及,由于前述内容已经相当复杂,度量学习较为小众,且和深度学习结合较紧,本书不对其做过多介绍。

10.2 排序学习

排序学习(learning to rank)在信息检索中常见,也广泛应用于其他领域。在一个排序学习的训练中,共会有N个训练样例,每一个训练样例X_i都包含了若干r_i个需要排序的项(通常为文档或图片)及其对应的得分。排序学习的目标是学习得到一个打分函数f,使得在测试时对任意一组待排序的项目,根据打分函数f的结果就能得到正确的排序。

对$i=1,\dots,N$,训练样例X_i可以表示为一个有序数组:$X_i=\{(\boldsymbol{x}_{i,j},y_{i,j})\}_{j=1}^{r_i}$。其中$\boldsymbol{x}_{i,j}$是第一个训练样例中$j=1,\dots,r_i$的第几个项,$\boldsymbol{x}_{i,j}^{(1)}$是第一个需要排序的项目;$\boldsymbol{x}_{i,j}^{(2)}$是与第一个项目对应的需要排序的项目;

$x_{i,j}^{(3)}$□□□□□,□□□□□□$y_{i,j}$□□□□(1,2,…,r_i),□□□□□□□□,□□□□□□□□
□□□□

□□□□□3□□□:□□□(pointwise)□□□(pairwise)□□□□
(listwise)□

□□□□□□□□□□□□□□□:□□□□□□□□□□□,□□□□□□□□□□□,□
□□□□□□□□□□□□□□□□□□(**x**,*y*)□,**x**□□□□□□□,*y*□□□□(□□$y_{i,j}$□□
□□),□□□□□□□□□□(□□□,□□□□□)□□□□,□□□□□□□□□□□□,□□
□□□□□□□□,□□□□□□□□□,□□□(□□□□□□□□$y_{i,j}$□□)□□□□□□□□
□□□□,□□□□□□,□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□

□□□□□□,□□□□□□□,□□□□□□□□□□□□($\boldsymbol{x}_i$,$\boldsymbol{x}_k$),□□□□□□□*f*□
($\boldsymbol{x}_i$,$\boldsymbol{x}_k$)□□□□□□□□□$\boldsymbol{x}_i$□□□□□$\boldsymbol{x}_k$□□,□□□□□1;□□,□□□□□0□□□
□,□□□□□□*X*□□□□□□□*X*□□□□□□□□□□□□□□□□□□□□□□,□□□
□□□□□

□□□□□□□□(□**LambdaMART**)□□□□□□□□□□□□,□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□
□□□□□□□□□□□(Mean Average Precision,MAP)□

□□MAP,□□□□□□□[□□□□□□□□(ranker)]□□□□□□□□□□□□
□□□,□□□□□□□□□□□□□□□□□(1□“□□”,0□“□□”),□□□□□□,□□
1□5:□□□□,□□□□□□□□□□□□□□□□100□□□□□□□□□□□□□,□□
□□□□□□□□□□□□□□□□□□□□□:

$$精确度 = \frac{|\{相关文档\} \cap \{搜索返回文档\}|}{|\{搜索返回文档\}|}$$

□□,$|\cdot|$□□□□□□□□□□□□□□□□□□□□□□□□□□q□□□□□□□□□□□□
(Average Precision,AveP)□□□□□□:

$$\mathrm{AveP}(q) = \frac{\sum_{k=1}^{n}(P(k) \cdot \mathrm{rel}(k))}{|\{相关文档\}|}$$

□□,n□□□□□□□□;$P(k)$□□□□□,□□□□□□□□□□k□□□□□□□;rel(k)□□□□□,□k□□□□□□□□□□□□(□□□□□□□)□□1,□□□0□□□,Q□□□□□□MAP□□□□:

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP}(q)}{Q}$$

□□,□□□□□□□LambdaMART□□□□□□□□□□□,□□□□□□□□□□□ $h(\boldsymbol{x})$□□□,□□□□□□□$f(\boldsymbol{x}_i,\boldsymbol{x}_k)$□□□□$\boldsymbol{x}_i$□□□$\boldsymbol{x}_k$□□□□(□□□□□□□□□)□□□□□□□□□□□□□□α□sigmoid□□□

$$f(\boldsymbol{x}_i, \boldsymbol{x}_k) \stackrel{\mathrm{def}}{=} \frac{1}{1 + \exp\left(h(\boldsymbol{x}_i) - h(\boldsymbol{x}_k)\right)\alpha}$$

□□,□□□□□□□□□□□□□□,□□□□□□□□□f□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□h□□□□□,□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□,□□□□□□□□□□□□□□□□□LambdaMART□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□MAP□□□□□□□□□□□□□□□□□□□□□

[illegible],[illegible],[illegible],[illegible](...)[illegible],[illegible],[illegible]LambdaMART[illegible]

[illegible],[illegible]f[illegible]f[illegible],[illegible],[illegible],[illegible]

[illegible](sorting)[illegible][[illegible](bubblesort)[illegible]][illegible],[illegible],[illegible]f[illegible],[illegible]

10.3 [illegible]

[illegible],[illegible],[illegible],[illegible](Netflix)[illegible],[illegible](Amazon)[illegible]

[illegible]:[illegible](content-based filtering)[illegible](collaborative filtering)[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible] $\boldsymbol{x}$,[illegible] y [illegible],[illegible],[illegible]

[illegible],[illegible]:[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible],[illegible]:[illegible],[illegible]10[illegible],[illegible]1[illegible]2[illegible],[illegible],[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]([illegible]0)[illegible],[illegible]

[illegible]:[illegible]

[illegible],[illegible]([illegible])[illegible]

[illegible]:[illegible](Factorization Machines,FM)[illegible](Denoising Autoencoders,DAE)[illegible]

10.3.1 [illegible]

[illegible],[illegible]

[illegible]10.1[illegible],[illegible],[illegible]x_{99}[illegible]x_{100}[illegible]y[illegible]

	用户				电影					电影评价								
	Ed	Al	Zak	…	It	Up	Jaws	Her		It	Up	Jaws	Her					
	x_1	x_2	x_3	…	x_{21}	x_{22}	x_{23}	x_{24}	…	x_{40}	x_{41}	x_{42}	x_{43}	…	x_{99}	x_{100}	y	
$\boldsymbol{x}^{(1)}$	1	0	0	…	1	0	0	0	…	0.2	0.8	0.4	0	…	0.3	0.8	1	$y^{(1)}$
$\boldsymbol{x}^{(2)}$	1	0	0	…	0	1	0	0	…	0.2	0.8	0.4	0	…	0.3	0.8	3	$y^{(2)}$
$\boldsymbol{x}^{(3)}$	1	0	0	…	0	0	1	0	…	0.2	0.8	0.4	0	…	0.3	0.8	2	$y^{(3)}$
$\boldsymbol{x}^{(4)}$	0	1	0	…	0	0	1	0	…	0	0	0.7	0.1	…	0.35	0.78	3	$y^{(4)}$
$\boldsymbol{x}^{(5)}$	0	1	0	…	0	0	0	1	…	0	0	0.7	0.1	…	0.35	0.78	1	$y^{(5)}$
$\boldsymbol{x}^{(6)}$	0	0	1	…	1	0	0	0	…	0.8	0	0	0.6	…	0.5	0.77	4	$y^{(6)}$
⋮	⋮	⋮	⋮												⋮	⋮	⋮	⋮
$\boldsymbol{x}^{(D)}$	0	0	0	…	0	0	1	0	…	0	0	1	0	…	0.95	0.85	5	$y^{(D)}$

[illegible]10.1 [illegible]$\boldsymbol{x}$[illegible]y

[illegible],[illegible],[illegible]10.1[illegible],[illegible],[illegible]x_{99}[illegible]x_{100}[illegible],[illegible]

[illegible],[illegible]

因子分解机的模型公式为:

$$f(\boldsymbol{x}) \stackrel{\text{def}}{=} b + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=i+1}^{D} (\boldsymbol{v}_i \boldsymbol{v}_j) x_i x_j$$

其中,b和w_i(i=1,...,D)是标量参数,与线性回归中使用的参数相似。$\boldsymbol{v}_i$是由因子组成k维向量(factor)。其中k是超参数,通常比D小得多。$\boldsymbol{v}_i\boldsymbol{v}_j$表示第$i$个与第$j$个因子向量的点积。

可见,因子分解机不是寻找一个宽泛的,数量很多的参数,而是通过添加一个附加项来捕获特征与特征之间的成对交互。例如$x_i x_j$的交互项有自己的参数,与使用特征对的单独参数$w_{i,j}$,该参数可能有数十亿个参数[1]不同,因子分解模型,其参数可以被减少为$Dk<<D(D-1)$个参数[2],而不是$w_{i,j}$我们用$\boldsymbol{v}_i\boldsymbol{v}_j$。

根据问题的不同,损失函数可以是平方误差损失(回归)。例如对于二分类y∈{-1,+1}的分类问题,可以使用逻辑损失函数。二分类模型的预测值为y=sign(f(x)),逻辑损失函数定义为:

$$\text{loss}(f(\boldsymbol{x}), y) = \frac{1}{\ln 2} \ln\left(1 + \mathrm{e}^{-y f(\boldsymbol{x})}\right)$$

利用梯度下降法来优化平均损失。在图10.1的例子中,标签属于{1,2,3,4,5},因此它是一个多类问题。我们可以使用一对其余(one versus rest)策略,将多类问题转换为5个二分类问题。

10.3.2　降噪自编码器

第7章中介绍了瓶颈层:在自编码器的训练中,瓶颈层的维数比输入低,同时作为输入的嵌入,在瓶颈层之前的网络称为编码器,之后的网络称为解码器的重构输入。

降噪自编码器:被训练为从部分被破坏的输入中重构原始输入的自编码器叫作降噪自编码器,在训练中随机将输入的一部分值,来破坏输入的完整性。

[illegible],[illegible]10.1[illegible],[illegible]

[illegible],[illegible],[illegible]0[illegible]0[illegible]

[illegible],[illegible],[illegible],[illegible]x_{99}[illegible]x_{100}[illegible]*DAE*[illegible],[illegible]

[illegible]*FFNN*[illegible]8[illegible],[illegible],[illegible]($\boldsymbol{u}$,$\boldsymbol{m}$,r)[illegible]$\boldsymbol{u}$[illegible]$\boldsymbol{m}$[illegible]r[illegible][0,1][illegible],[illegible]sigmoid;[illegible],[illegible][1,5][illegible],[illegible]ReLu[illegible]

10.4 [illegible]:[illegible]

[illegible]7[illegible](word embedding)[illegible],[illegible],[illegible]?[illegible]([illegible])[illegible]:[illegible]

[illegible],[illegible]word2vec,[illegible]word2vec[illegible](skip-gram)[illegible]word2vec

□□□

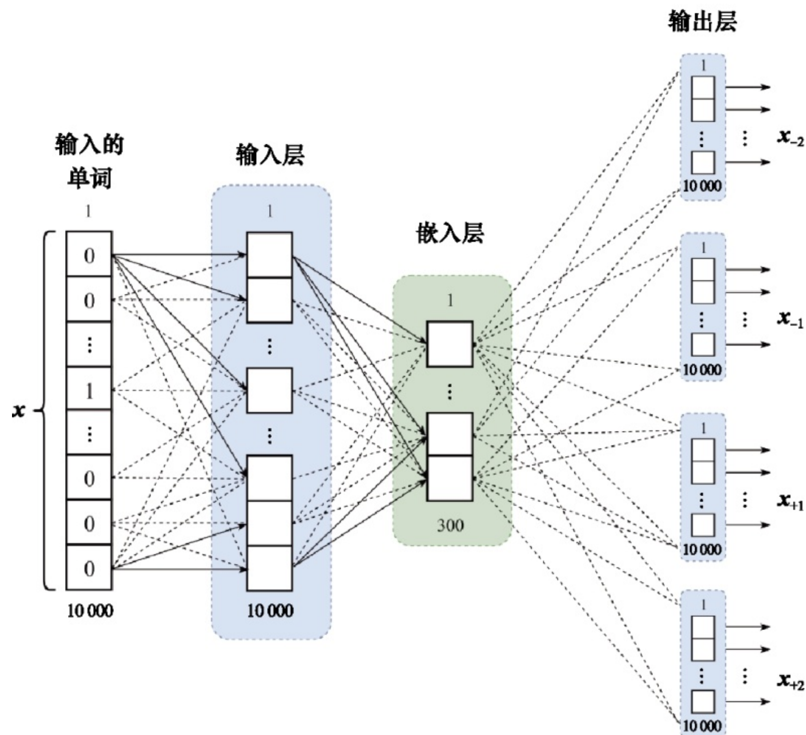
□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□10 000□□□□□□□□□□□□□□10 000□□□,□□1□□□□□□1□□,□□□□□□□□□0□

□□□□□□“I almost finished reading the book on machine learning.”(□□□□□□□□□□□□□□□)□□[3],□□□□□□,□□“book”(□),□□□□□□,□□□□□□□□□“I almost finished reading the • on machine learning.”□□□,□□□□□•□□□□□□□□3□□□“finished reading the ? on machine learning”□□□□□□□□□7□□□□□□,□□□□“?”□□□□,□□□□□□“book”(□)“article”(□□)□“paper”(□□)□□□□□□□□□□□□□□□□□□□□□“book”“article”□“paper”□□□□□□□□□:□□□□□□□□□□□□□□□

□□□,□□□□□□□□□□□□□□□□□□□□“finished reading the•on machine learning”□□□□□□□□□□7(3+1+3)□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□

□□□□$[\boldsymbol{x}_{-3},\boldsymbol{x}_{-2},\boldsymbol{x}_{-1},\boldsymbol{x},\boldsymbol{x}_{+1},\boldsymbol{x}_{+2},\boldsymbol{x}_{+3}]$□□□□□□□□□□□□□□,$\boldsymbol{x}_{-3}$□□□“finished”(□□)□□□□□,$\boldsymbol{x}_{-2}$□□“reading”(□),$\boldsymbol{x}$□□□•,$\boldsymbol{x}_{+1}$□“on”,□□□□□□□□□□□□□□5□□□□□□□□□□□□:$[\boldsymbol{x}_{-2},\boldsymbol{x}_{-1},\boldsymbol{x},\boldsymbol{x}_{+1},\boldsymbol{x}_{+2}]$□

□10.2□□□□□□□□□□□5□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□•□□□□□□□□□□□□□□□□□□□□□□□□



□10.2 □□□□□□□5□□□□□□□□300□□□□□

□□□□□□□□□□□□□□□□□□□□(self-supervised):□□□□□□□□□□□□
□□□□□□,□□□□



□□□□□□□□□□□softmax□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□
□□□□□□□□□□□□□

word2vec[illegible],[illegible]:[illegible]softmax(hierarchicalsoftmax)([illegible]softmax[illegible],[illegible]softmax[illegible])[illegible](negativesampling)([illegible],[illegible])[illegible]

---

[1] [illegible],[illegible]$D(D-1)$[illegible]$w_{i,j}$[illegible]

[2] <<[illegible]

[3] [illegible]:[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible]11[illegible] [illegible]

[illegible],[illegible]![illegible],[illegible]

[illegible],[illegible],[illegible],[illegible]:[illegible],[illegible]

[illegible],[illegible]?

11.1 [illegible]

[illegible],[illegible]:[illegible],[illegible](Latent Dirichlet Allocation,LDA)[illegible],[illegible],[illegible],[illegible],[illegible]

11.2 [illegible]

[illegible](Gaussian Process,GP)[illegible],[illegible],[illegible],[illegible],[illegible]

11.3 [illegible]

[illegible](Generalized Linear Model,GLM)[illegible],[illegible],[illegible]GLM[illegible],[illegible],[illegible]GLM[illegible]

11.4 [illegible]

[illegible],[illegible]7[illegible](Probabilistic Graphical Model,PGM):[illegible](Conditional Random Field,CRF)[illegible]CRF[illegible](dependencygraph)[illegible],[illegible]PGM[illegible](graph)[illegible]([illegible]),[illegible],[illegible]“[illegible]”[illegible]“[illegible]”[illegible],[illegible]

PGM[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]CRF[illegible]PGM,[illegible],[illegible],CRF[illegible](Hidden Markov Model,HMM),[illegible],HMM[illegible]

[illegible]PGM,[illegible](Bayesion network)[illegible](belief network)[illegible](probabilistic independence network)[illegible]

11.5 [illegible]

[illegible],[illegible],[illegible](Markov Chain Monte Carlo,MCMC)[illegible]MCMC[illegible],[illegible],[illegible](denoising autoencoder)[illegible],[illegible]([illegible])[illegible],[illegible],[illegible],[illegible]

11.6 [illegible]

[illegible](Genetic Algorithm,GA)[illegible],[illegible]([illegible])[illegible]

[illegible],GA[illegible],[illegible],[illegible],[illegible]“[illegible]”“[illegible]”[illegible]“[illegible]”[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]“[illegible]”[illegible]“[illegible]”[illegible],[illegible]([illegible])[illegible]

[illegible],GA[illegible],[illegible]

11.7 [illegible]

[illegible],[illegible](Reinforcement Learning,RL)[illegible],[illegible],[illegible](agent)[illegible](reward),[illegible]([illegible])[illegible]

[illegible],[illegible]Q[illegible](Q-learning),[illegible],[illegible]([illegible])[illegible]

[illegible],[illegible],[illegible],[illegible],[illegible]

[illegible]

A

accuracy([illegible]) [illegible],[illegible]

activation function([illegible]) [illegible],[illegible],[illegible]

active learning([illegible]) [illegible],[illegible],[illegible],[illegible]

AdaBoost[illegible]AdaBoost[illegible],[illegible],[illegible],[illegible]

agglomerative clustering algorithm([illegible]) [illegible]"[illegible]"[illegible],[illegible]

area under the ROC curve(**ROC**[illegible]) [illegible]ROC[illegible]([illegible],[illegible])[illegible]([illegible]),[illegible]ROC[illegible],[illegible],[illegible]

autoencoder([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

B

backpropagation([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

bag of words([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible]

bagging([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible]([illegible])[illegible]

base learner([illegible]) [illegible],[illegible],[illegible]

baseline([illegible]) [illegible](heuristic)[illegible]

batch([illegible]) [illegible](minibatch),[illegible]

batch normalization([illegible]) [illegible],[illegible]0[illegible]([illegible]:[illegible])[illegible],[illegible],[illegible],[illegible](regularization)[illegible]

Bayes’ Rule([illegible]) [illegible],[illegible]18[illegible]•[illegible],[illegible]

Bayesian optimization([illegible]) [illegible],[illegible]

bias([illegible]) [illegible]([illegible])[illegible]

bias-variance tradeoff([illegible]-[illegible]) [illegible],[illegible]([illegible])[illegible]([illegible])[illegible]-[illegible]

bigram([illegible]([illegible])[illegible]) [illegible],[illegible]“cat”[illegible],[illegible]“ca”[illegible]“at”[illegible]“[illegible]”[illegible]3[illegible]:“[illegible]”“[illegible]”[illegible]“[illegible]”[illegible],[illegible],[illegible]

bin([illegible]) [illegible],[illegible]1[illegible]

binary classification([illegible]) [illegible],[illegible],[illegible]

binary variable([illegible]) [illegible]([illegible]):1[illegible]0,“[illegible]”[illegible]“[illegible]”[illegible]

binning([illegible]) [illegible],[illegible]([illegible]),[illegible],[illegible],[illegible]:[illegible]0~5[illegible],6~10[illegible],11~15[illegible],16~25[illegible],[illegible]

boosting([illegible]) [illegible],[illegible]([illegible])[illegible]([illegible])[illegible],[illegible],[illegible]AdaBoost[illegible]

bootstrap aggregating([illegible]) [illegible]Bagging([illegible])[illegible]

Bootstrapping([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

Bottleneck([illegible]) [illegible],[illegible],[illegible],[illegible]

bucket([illegible]) [illegible]bin([illegible])[illegible]

bucketing([illegible]) [illegible]binning([illegible])[illegible]

C

C4.5[illegible]ID3,[illegible]:[illegible],[illegible]“[illegible]”[illegible]

CART(□□□□□□)　□□□□□□□□□□□□,□C4.5□□□

categorical feature(□□□□)　□□□□□□□□□□□□□□

categorical variable(□□□□)　□□□□□□□□□□□,□□□□□□□□□□□□,□□□□“□□□□□”□□□3□□□□:“□”“□”□“□”□

centroid(□□)　k□□□k□□□□□□□□□□□□□□□□□□□,□□*k*□□4,k□□□k□□□□□□□4□□□□

class(□□)　□□,□□□□□□□□□□□□□□,□□□□□□□“□□□□”□“□□□□□”□□□□□□□□□□□□□□□,□□□□□“□”“□”“□□”□□

classification(□□)　□□□□□□□□□□□□□

classification algorithm(□□□□)　□□□□□□□□,□□□□□□□□□□□□□□□□□□□□

classification model(□□□□)　□classification(□□)□

classification threshold(□□□□)　□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□:□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□0.9,□□□□□□0.9□□□□□□□,□□0.9□□□□□□□

classifier(□□□)　□model,classification(□□,□□)□

Cluster(□□□)　□□□□□□□□□□□□□□□□(□□□□□□□□□□□□)□

Clustering([illegible]) [illegible]

clustering algorithm([illegible]) [illegible],[illegible],[illegible],[illegible]k[illegible]DBSCAN[illegible]

computational graph([illegible]) [illegible],[illegible],[illegible],[illegible]

confusion matrix([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible],[illegible]“[illegible]”[illegible]“[illegible]”,[illegible]“[illegible]”[illegible]“[illegible]”[illegible],[illegible]

continuous variable([illegible]) [illegible],[illegible]

convergence([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible],[illegible]

convolution([illegible]) [illegible]

[illegible],[illegible],[illegible]

convolution filter([illegible]) [illegible],[illegible]

convolutional neural network(**CNN**,[illegible]) [illegible],[illegible],CNN[illegible],[illegible],[illegible]

cosine similarity([illegible]) [illegible],[illegible]

cost function([illegible]) [illegible],[illegible],[illegible]([illegible])[illegible]

cross entropy([illegible]) [illegible]X[illegible]p[illegible]q,[illegible]$H(p,q)=-\sum_{x\in X}p(x)\log q(x)$[illegible],[illegible]

cross-validation([illegible]) [illegible],[illegible],[illegible]

D

dataset([illegible]) [illegible]

DBSCAN [illegible],[illegible]

decision boundary([illegible]) [illegible],[illegible]
[illegible],[illegible]

decision tree([illegible]) [illegible],[illegible]([illegible])[illegible]([illegible])[illegible]ID3[illegible]C4.5[illegible]
CART[illegible]

decision tree learning algorithm([illegible]) [illegible]ID3[illegible]CART[illegible]C4.5[illegible]

deep neural network([illegible]) [illegible]

density-based clustering algorithm([illegible]) [illegible],[illegible]([illegible]),[illegible],[illegible]

dimensionality reduction([illegible]) [illegible],[illegible]([illegible])[illegible](Principal Component Analysis,PCA)[illegible]UMAP[illegible],[illegible]
word2vec[illegible],[illegible],[illegible],[illegible]

divisive clustering algorithm([illegible]) [illegible]:[illegible],[illegible],[illegible]

dropout([illegible]) [illegible],[illegible]0,[illegible]

dummy variable([illegible]) [illegible]0[illegible]1[illegible],0[illegible]25,1[illegible]25[illegible]:binning([illegible])[illegible]

E

early stopping([illegible]) [illegible],[illegible],[illegible]

embedding([illegible]) [illegible],[illegible],[illegible]([illegible]),[illegible]word2vec[illegible]GloVe[illegible],[illegible],[illegible],[illegible],[illegible]

ensemble learning([illegible]) [illegible]

ensemble learning algorithm([illegible]) [illegible]([illegible])[illegible]

epoch([illegible]) [illegible]

evaluation metric([illegible]) [illegible],[illegible]ROC[illegible]F[illegible]

evolutionary algorithm([illegible]) [illegible],[illegible]“[illegible]”“[illegible]”[illegible]

[illegible]

evolutionary optimization([illegible]) [illegible],[illegible]

example([illegible]) [illegible],[illegible],[illegible],[illegible],[illegible]([illegible]),[illegible]([illegible]),[illegible],[illegible],[illegible]

example,**labeled**([illegible],[illegible]) [illegible],[illegible],[illegible],labeled example([illegible])[illegible]

example,**training**([illegible],[illegible]) [illegible]

example,**unlabeled**([illegible],[illegible]) [illegible],[illegible],unlabelled example([illegible])[illegible]

exploding gradient problem([illegible]) [illegible],[illegible],[illegible]

F

F score(**F**[illegible]) [illegible]F1 score(F1[illegible])[illegible]

F1 score(**F1**[illegible]) [illegible]F[illegible]F[illegible],[illegible],[illegible]

false positive(**FP**,)

feature()

feature engineering()

feature selection()

feature vector()

feature,**informative**(,)

feedforward neural network(**FFNN**,)

few-shot learning()

G

gate()

gated recurrent unit(**GRU**,) GRU

□□□□□□□□□□□□□GRU□□□□□□□(□□□□□□□□□□□□□□)□□□□□□□(□□□□□□□□□□□□□)□

gated unit(□□□)　□□□□□□□□□□□□□□□□,□□□□“□□□”□□□□□□□
□□□□□□□□□□□□□□□□(GRU)□□□□□□□(LSTM)□

Gaussian distribution(□□□□)　□□□□□□,□□□□□□□□□□□□□
□□□□□□□□□□$f_{\mu,\sigma^2}(x)=\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$,□□$\mu$□□□□□□(□□□□),$\sigma^2$□□□□

Gaussian process(□□□□)　□□□□□□□□□,□□□□□□□□□□□□□□□□□

generalization(□□□)　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

genetic algorithm(**GA**,□□□□)　□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□)□□□□□□,GA□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□“□□”“□□”□“□□”□□□□□□□□*d*□□□□□□□□

GloVe(□□□□)　□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□word2vec□

gradient boosting(□□□□)　□□□□□□□□,□□□

gradient clipping([illegible]) [illegible],[illegible],[illegible]

gradient descent([illegible]) [illegible],[illegible],[illegible]:[illegible](SGD)[illegible]

gradient step([illegible]) [illegible],[illegible]

grid search([illegible]) [illegible],[illegible]

H

hidden layer([illegible]) [illegible],[illegible]

hidden unit([illegible]) [illegible],[illegible]

hierarchical clustering algorithm([illegible]) [illegible]:[illegible]:[illegible]

hinge loss([illegible]) [illegible],[illegible]

holdout set([illegible]) [illegible],[illegible](“[illegible]”)[illegible],[illegible]

hyperparameter([illegible]) [illegible],[illegible],[illegible],[illegible]:[illegible]

hyperparameter tuning([illegible]) [illegible],[illegible],[illegible]

Hyperplane([illegible]) [illegible],[illegible];[illegible],[illegible],[illegible],[illegible]

I

ID3[illegible]

input layer([illegible]) [illegible],[illegible]

instance([illegible]) [illegible]

instance-based learning([illegible]) [illegible],[illegible],[illegible],[illegible]k[illegible]

Iteration([illegible]) [illegible],[illegible],[illegible]

K

k Nearest Neighbor(kNN)(k□□)　□□□□□□□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□k□□□□□□□□□□□□□□□□□□□□□□□□□□,□□k□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

k-mean(k-□□)　□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□k□□□□□□□□□□□□□□□□□□□□□□□□

k-median(k-□□)　□□□□□□□□,□□□□□□□□□□□□□□,□□□□□□□k□□□□□□□□□□□□□□□□□□□□□□□□

Kernel(□)　□□□□□□x'□□□□□□□x_j□□□□□□□□□□□□□□□□□□□□□□□□□

kernel trick(□□□)　□□□□□□□□□x□x'□□□,□□□□□□□□□$\varphi(x)$□$\varphi(x')$□□□□□□□□□□□□□□□□□□□□□□□□□□□□

L

L1 regularization(L1□□□)　□□□□□□□,□□□□□□□□□□□□□□□□□

L2 regularization(L2□□□)　□□□□□□□,□□□□□□□□□□□□□□□□

label(□□)　□□□□□□□,□□□□□□□□□□□□(□□□)□□□□(□□□)□

labeled example(□□□□□)　□example,labeled(□□,□□□)□

Latent Dirichlet Allocation(**LDA**,[illegible]) [illegible],[illegible]

latent semantic Indexing([illegible]) [illegible],[illegible]

layer([illegible]) [illegible],[illegible]

learning algorithm([illegible]) [illegible],[illegible]

learning rate([illegible]) [illegible],[illegible],[illegible]

linear regression([illegible]) [illegible],[illegible]

log loss([illegible]) [illegible]

logistic regression([illegible]) [illegible],[illegible],[illegible],[illegible]sigmoid[illegible]

“[illegible]”[illegible],[illegible]“[illegible]”[illegible]

long short-term memory(**LSTM**)**unit**([illegible]) [illegible],[illegible],LSTM[illegible]LSTM[illegible]3[illegible](“[illegible]”“[illegible]”[illegible]“[illegible]”)[illegible]RNN[illegible],[illegible]LSTM[illegible]LSTM[illegible],[illegible]RNN[illegible]RNN[illegible]

loss([illegible]) [illegible]

loss function([illegible]) [illegible],[illegible]:[illegible],[illegible];[illegible]

LSTM network([illegible]) [illegible]LSTM[illegible]

M

machine learning([illegible]) [illegible],[illegible],[illegible],[illegible]:[illegible],[illegible]

machine learning algorithm([illegible]) [illegible]machine learning([illegible])[illegible]

maximum margin classification([illegible]) [illegible],[illegible],[illegible]

metric learning([illegible]) [illegible],[illegible],[illegible]“[illegible]”:[illegible],[illegible],[illegible]

minibatch([illegible]) [illegible]batch([illegible])[illegible]

minibatch stochastic gradient descent([illegible]) [illegible],[illegible],[illegible]“[illegible]”[illegible]

model(□□)　□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□

model,**classification**(□□,□□)　□□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□

model,**regression**(□□,□□)　□□□□□□□□□□,□□□□□□□□□□□

multi-class classification(□□□□□)　□□□□□□□□□□□□□□□□□□,□□□“□□”“□□”“□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□,□□□□□□□(“□□□□”□“□□□□□”)□□□□□□□□□□□□

multi-class logistic regression(□□□□□□□□□)　□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□□□

N

neural network(□□□□)　□□□□□□□□(□□□□□□)□□□□□□□□

neuron(□□□)　□unit(□□)□

normalization(□□□)　□□□□□□□□□□□□□□□□□□□,□□□□□□□□□[-1,+1]□[0,1]□

O

objective function([illegible]) [illegible],[illegible]

one-hot encoding([illegible]) [illegible],[illegible]1,[illegible]0[illegible]

outlier([illegible]) [illegible]

output layer([illegible]) [illegible],[illegible]

output unit([illegible]) [illegible]

overfitting([illegible]) [illegible],[illegible],[illegible]

P

parameter([illegible]) [illegible],[illegible],[illegible],[illegible]

partitional clustering algorithm([illegible]) [illegible],[illegible],[illegible]k[illegible]k[illegible]

pooling([illegible]) [illegible]

population([illegible]) [illegible],[illegible]([illegible])[illegible],[illegible],[illegible]

precision([illegible]) [illegible],[illegible]:recall([illegible])[illegible]confusion matrix([illegible])[illegible]

predictive model([illegible]) [illegible]

predictive power([illegible]) [illegible]

principal component analysis(**PCA**,[illegible]) [illegible]PCA[illegible],[illegible]

R

random forests([illegible]) [illegible],[illegible]

random search([illegible]) [illegible],[illegible]

random variable([illegible]) [illegible],[illegible],[illegible]

recall([illegible]) [illegible],[illegible]:precision([illegible])[illegible]confusion matrix([illegible])[illegible]

recurrent neural network(**RNN**,[illegible]) [illegible],[illegible],[illegible]RNN[illegible],[illegible]

regression([illegible]) [illegible],[illegible]([illegible])[illegible]

regression algorithm([illegible]) [illegible]

regression model([illegible]) [illegible]model,regression([illegible],[illegible])[illegible]

regularization([illegible]) [illegible],[illegible]
[illegible]L1[illegible]L2[illegible]

reinforcement learning([illegible]) [illegible],[illegible]
[illegible],[illegible],[illegible]
[illegible]

representation learning([illegible]) [illegible],[illegible]
[illegible]

risk([illegible]) [illegible]

S

semi-supervised learning([illegible]) [illegible]
[illegible],[illegible]

sigmoid function(**S**[illegible]) [illegible](0,1)[illegible]
[illegible]

similarity metric([illegible]) [illegible],[illegible],[illegible]
[illegible],[illegible],[illegible]

standardization(□□□)　□□z□□□,□□□□□□□□□□□□□□□□μ=0□σ=1□□□□□□□□□□□□□,μ□□□(□□□□□□□□□□□□□□□□),σ□□□□□

statistic(□□)　□□□□□□□□□,□□□□□□□□

statistical model(□□□□)　□model(□□)□

stochastic gradient descent(**SGD**,□□□□□□)　□□□□□□□□□□,□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□,□□□SGD□□□□□

strong classifier(□□□□)　□□□

supervised learning(□□□□)　□□□□□□□□□□□□□□□□□□□□□□□□

Support Vector Machine(**SVM**,□□□□□)　□□□□□□,□□□□□□□□□□□□□□□□□□SVM□□□□□□□□□□□□□

T

target(□□)　□□□□□□□□□□□□□□□□□□□□

test set(□□□)　□□□□□□□□□□□□□□□□□:training set(□□□)□validation set(□□□)□

token(□□)　□□□□□□□□□□□,□□□□□□□□

tokenization(□□□) □□□□□□□,□□□□□□□□□□□□□
□□□□□□□□□□

topic modelling(□□□□) □□□□□□□□□□□,□□□□□□
□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□
(LDA)□□□□(LSI)□

training(□□) □□□□□□□□□□□□□□,□□□□□□□

training data(□□□□) □traing set(□□□)□

training example(□□□□) □example,training(□□,□□)□

training loss(□□□□) □□□□□□□□□□□□□□□□□

training set(□□□) □□□□,□□□□□□□,□□□□□□□□
□:test set(□□□)□validation set(□□□)□

transfer learning(□□□□) □□□□□,□□□□□□□□□□□□
□□□□□

true negative(**TN**,□□) □□□□□,□□□□□□□□□□□□□

true positive(**TP**,□□) □□□□□,□□□□□□□□□□□□□

U

UMAP(□□□□□□□□□)　□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□

underfitting(□□□)　□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□
□□□□□

unit(□□)　□□□□□□□□□,□□□□□□□□□,□□□□□□□□□□,□□□□□□□□
□□□□□□□□□□□□□□□(□□□□□)□□□□□□□□□□□□□□□□□

unlabeled example(□□□□□)　□example,unlabeled(□□,□
□□)□

unsupervised learning(□□□□□)　□□□□□□□□□□□□□□□□(□
□□)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

unsupervised learning algorithm(□□□□□□□)　□□□□□□
□□□□□□□□□

V

validation example(□□□□)　□□□□□□□

validation loss(□□□□)　□□□□□□□□□□□□□□□□□□□□□□□□

validation set(□□□)　□□□□□□□□□□□□□

vanishing gradient problem(□□□□□□)　□□□□□□□□□□□□
□□□□□,□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□□,□□

[illegible]“[illegible]”,[illegible]

variance([illegible]) [illegible]
[illegible],[illegible]([illegible])[illegible]

volume([illegible]) [illegible],[illegible]
[illegible],[illegible](CNN)[illegible]

W

weak classifier([illegible]) [illegible],[illegible]
[illegible],[illegible]

word embedding([illegible]) [illegible];[illegible]
[illegible],[illegible],[illegible]

word2vec [illegible],[illegible],[illegible],[illegible]
[illegible]:GloVe([illegible])[illegible]

Z

zero shot learning([illegible]) [illegible],[illegible]
[illegible],[illegible]